Python Exercise Problem Statements

1. 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.

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

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

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

5. Create a **Bus** child class that inherits from the Vehicle class. The default fare charge of any vehicle is **seating capacity * 100**. If Vehicle is **Bus** instance, we need to add an extra 10% on full fare as a maintenance charge. So total fare for bus instance will become the **final amount = total fare + 10% of the total fare.**

6. Define a **class** attribute "**color**" with a default value **white**. i.e., Every Vehicle should be white.

7. Implement a class inheritance as following:

```
class Shoe:
   # Attributes: self.color, self.brand

class Converse(Shoe): # Inherits from Shoe
   # Attributes: self.lowOrHighTop, self.tongueColor, self.brand = "Converse"

class CombatBoot(Shoe): # Inherits from Shoe
   # Attributes: self.militaryBranch, self.DesertOrJungle

class Sandal(Shoe): # Inherits from Shoe
   # Attributes: self.openOrClosedToe, self.waterproof
```

   You can use any real-world object *except a shoe* for this problem :)

8. Create a Python class called "Car" with attributes like make, model, and year. Then, create an object of the "Car" class and print its details.

9. Create a base class called “Animal” and two subclasses, “Dog” and “Cat.” Add methods and attributes specific to each subclass.

10. Create three classes, “Person,” “Employee,” and “Student.” Use multiple inheritance to create a class “PersonInfo” that inherits from both “Employee” and “Student.” Add attributes and methods specific to each class.

11. Create a base class called “Vehicle” with a method called “drive.” Implement two subclasses, “Car” and “Bicycle,” that inherit from “Vehicle” and override the “drive” method with their own implementations.

12. Define a class with a generator which can iterate the numbers, which are divisible by 7, between a given range 0 and n.

13. Write a program to compute the frequency of the words from the input. The output should output after sorting the key alphanumerically.

14. To find the Euclidean distance between two points in a two dimensional space using class and object

15. Create a Python program for an online quiz system. Implement classes for quizzes, questions, and users. Include methods for taking quizzes, scoring, and displaying results.

16. Build a hotel reservation system with classes for rooms, guests, and reservations. Implement methods for checking room availability, booking rooms, and generating invoices.

17. Develop a time tracking system for employees with classes for employees, projects, and time entries. Implement methods for logging hours, generating timesheets, and calculating overtime.

18. Create a conference room booking system with classes for rooms, reservations, and users. Include methods for checking room availability, booking time slots, and sending notifications.

19. Design a recipe management system with classes for recipes, ingredients, and users. Implement methods for adding recipes, searching by ingredients

20. Build a simulation of an ATM system with classes for accounts, transactions, and users. Implement methods for withdrawing cash, checking balances, and handling PIN verification.