Exercises

1) Create a Movie class with title, director, and year attributes and a method to check if the movie was released before a given year.
2) Create a House class with address, bedrooms, and bathrooms attributes and a method to calculate the total square footage of the house (assuming each bedroom is 150 square feet and each bathroom is 50 square feet).
3) Create a Car class with make, model, year, and mileage attributes and a method to calculate the car's average mileage per year.
4) Create a Student class with name and grades attributes and a method to calculate the student's average grade.
5) Create a Rectangle class with height and width attributes and a method to check if the rectangle is a square and to calculate the diagonal length of the rectangle.
6) Create a BankAccount class with balance attribute and methods to deposit, withdraw, and transfer money to another account.
7) Create a Person class with name and age attributes and a method to change the person's name and age and to print out the person's new name and age.
8) Create a Dog class with name and breed attributes and a method to check if the dog is a puppy (age < 1 year).
9) Create a Cat class with name and color attributes and a method to check if the cat is a kitten (age < 1 year).
10) Create a Bank class with a list of BankAccount objects and methods to find the account with the highest balance, the average balance of all accounts, and the total number of accounts.

Exercises and solution

1) Movie class to check if the movie was released before a given year:

```
class Movie:
   def __init__(self, title, director, year):
      self.title = title
      self.director = director
      self.year = year

   def released_before_year(self, year):
      return self.year < year
```

2) House class to calculate the total square footage of the house:

```
class House:
   BEDROOM_AREA = 150
   BATHROOM_AREA = 50

   def __init__(self, address, bedrooms, bathrooms):
      self.address = address
      self.bedrooms = bedrooms
      self.bathrooms = bathrooms

   def total_square_footage(self):
      return self.bedrooms * House.BEDROOM_AREA + self.bathrooms *
House.BATHROOM_AREA
```

3) Car class to calculate the car's average mileage per year:

```
class Car:
   def __init__(self, make, model, year, mileage):
      self.make = make
      self.model = model
```

```
        self.year = year
        self.mileage = mileage

    def avg_mileage_per_year(self):
        current_year = 2023
        age = current_year - self.year
        return self.mileage / age if age > 0 else 0
```

4) Student class to calculate the student's average grade:

```
class Student:
    def __init__(self, name, grades):
        self.name = name
        self.grades = grades

    def avg_grade(self):
        return sum(self.grades) / len(self.grades) if self.grades else 0
```

5) Rectangle class to check if the rectangle is a square and to calculate the diagonal length of the rectangle:

```
import math

class Rectangle:
    def __init__(self, height, width):
        self.height = height
        self.width = width

    def is_square(self):
        return self.height == self.width

    def diagonal_length(self):
        return math.sqrt(self.height ** 2 + self.width ** 2)
```

6) BankAccount class with methods to deposit, withdraw, and transfer money to another account:

```
class BankAccount:
    def __init__(self, balance):
        self.balance = balance

    def deposit(self, amount):
        self.balance += amount

    def withdraw(self, amount):
        if amount > self.balance:
            print("Insufficient balance.")
        else:
            self.balance -= amount

    def transfer(self, amount, other_account):
        if amount > self.balance:
            print("Insufficient balance.")
        else:
            self.balance -= amount
            other_account.deposit(amount)
```

7) Person class with methods to change the person's name and age and to print out the person's new name and age:

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def change_name(self, new_name):
        self.name = new_name

    def change_age(self, new_age):
        self.age = new_age
```

```
  def print_info(self):
    print(f"Name: {self.name}, Age: {self.age}")
```

8) Dog class to check if the dog is a puppy (age < 1 year):

```
class Dog:
  def __init__(self, name, breed, age):
    self.name = name
    self.breed = breed
    self.age = age

  def is_puppy(self):
    return self.age < 1
```

9) Create a Cat class with name and color attributes and a method to check if the cat is a kitten (age < 1 year).

```
class Cat:
  def __init__(self, name, color, age):
    self.name = name
    self.color = color
    self.age = age

  def is_kitten(self):
    return self.age < 1
```

10) Create a Bank class with a list of BankAccount objects and methods to find the account with the highest balance, the average balance of all accounts, and the total number of accounts.

```
class Bank:
  def __init__(self):
    self.accounts = []

  def add_account(self, account):
    self.accounts.append(account)
```

```
def remove_account(self, account):
    self.accounts.remove(account)

def highest_balance(self):
    return max(self.accounts, key=lambda account: account.balance)

def average_balance(self):
    total_balance = sum(account.balance for account in self.accounts)
    return total_balance / len(self.accounts)

def total_accounts(self):
    return len(self.accounts)
```