Exercise 1:

1. Create a class called "Car" with the following properties:
   - private string "brand"
   - private int "year"
   - private float "price"
2. Implement a constructor that takes the brand, year, and price as parameters and initializes the corresponding properties.
3. Create an instance of the "Car" class and print out the brand, year, and price of the car.

Exercise 2:

1. Create a class called "BankAccount" with the following properties:
   - private string "accountNumber"
   - private decimal "balance"
2. Implement a constructor that takes the account number as a parameter and initializes the "accountNumber" property. The "balance" should be set to 0 by default.
3. Create a method called "Deposit" that takes an amount (decimal) as a parameter and adds it to the balance.
4. Create a method called "Withdraw" that takes an amount (decimal) as a parameter and subtracts it from the balance.
5. Create an instance of the "BankAccount" class, perform a deposit of 1000 units, and then withdraw 500 units. Print out the final balance.

Exercise 3:

1. Create a class called "Rectangle" with the following properties:
    - private float "length"
    - private float "width"
2. Implement a constructor that takes the length and width as parameters and initializes the corresponding properties.
3. Create a method called "CalculateArea" that calculates and returns the area of the rectangle (length * width).
4. Create an instance of the "Rectangle" class, set the length to 4.5 and the width to 3.2. Call the "CalculateArea" method and print out the result.

Exercise 4:

1. Create a base class called "Shape" with a virtual method called "CalculateArea" that returns 0.
2. Create a derived class called "Circle" that inherits from "Shape" and overrides the "CalculateArea" method to calculate and return the area of a circle using the formula: pi * radius^2.
3. Create an instance of the "Circle" class with a radius of 5. Call the "CalculateArea" method and print out the result.

Exercise 5:

1. Create a class called "Student" with the following properties:
    - private string "name"
    - private int "age"
    - private string "major"
2. Implement a constructor that takes the name, age, and major as parameters and initializes the corresponding properties.
3. Create a method called "Introduce" that prints out a message introducing the student with their name, age, and major.
4. Create an instance of the "Student" class, set the name to "John Doe", age to 20, and major to "Computer Science". Call the "Introduce" method.

Exercise 6:

1. Create a generic class called "Stack<T>" that implements a stack data structure. The class should have the following methods:
    - "Push" method that adds an item of type T to the top of the stack.
    - "Pop" method that removes and returns the top item from the stack.
    - "Peek" method that returns the top item from the stack without removing it.
    - "IsEmpty" method that returns true if the stack is empty, false otherwise.
2. Create an instance of the "Stack<int>" class and perform various push and pop operations. Print out the remaining items in the stack after each operation.

Exercise 7:

1. Create a nested enum type called "DaysOfWeek" inside a class called "Calendar". The enum should have the following values: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday.
2. Create a method called "PrintWeekdays" inside the "Calendar" class that prints out all the weekdays (Monday to Friday) from the "DaysOfWeek" enum.
3. Create an instance of the "Calendar" class and call the "PrintWeekdays" method.

Exercise 8:

1. Create a struct called "Point2D" that represents a 2-dimensional point with x and y coordinates as float values.
2. Create a method called "CalculateDistance" inside another class that takes two "Point2D" objects as parameters and calculates the Euclidean distance between them using the distance formula: sqrt((x2 - x1)^2 + (y2 - y1)^2).
3. Create two instances of the "Point2D" struct and calculate the distance between them using the "CalculateDistance" method. Print out the result.