Part01

- **Problem:**
  Create an enum called Weekdays with values Monday = 1, Tuesday, Wednesday, Thursday, Friday. Write a program to print the name and value of each day.
- **Question:**
  Why is it recommended to explicitly assign values to enum members in some cases?

- **Problem:**
  Modify the Grades enum from the demo to use short as its underlying type and set F to -1. Write code to print all grade values and their names.
- **Question:**
  What happens if you assign a value to an enum member that exceeds the underlying type's range?

- **Problem:**
  Add a Department property to the Person class. Instantiate two Person objects and assign different departments. Print their details.
- **Question:**
  What is the purpose of the virtual keyword when used with properties?

- **Problem:**
  Extend the Child class to include a method DisplaySalary that uses the sealed Salary property. Instantiate the class and demonstrate the use of this method.
- **Question:**
  Why can’t you override a sealed property or method?

- **Problem:**
  Write a static method in the Utility class to calculate the perimeter of a rectangle. Call it without creating an instance of the class.
- **Question:**
  What is the key difference between static and object members?

- **Problem:**
  Modify the ComplexNumber class to add operator overloading for the multiplication (*) operator. Demonstrate it with two complex numbers.
- **Question:**
  Can you overload all operators in C#? Explain why or why not.

- **Problem:**
  Modify the Gender enum to use byte as its underlying type. Write a program to demonstrate its memory usage compared to the default int.
- **Question:**
  When should you consider changing the underlying type of an enum?

- **Problem:**
  Create a static method in the Utility class to convert temperatures between Celsius and Fahrenheit. Write code to demonstrate its usage.
- **Question:**
  Why can't a static class have instance constructors?

- **Problem:**
  Write a program that tries to parse a string to a Grades enum value. Use Enum.TryParse to handle invalid inputs gracefully.
- **Question:**
  What are the advantages of using Enum.TryParse over direct parsing with int.Parse?

- **Problem:**
  Enhance the Employee class to include Equals method. Demonstrate the correct usage of Equals when searching for an employee object in an array using Helper2<Employee>.SearchArray.
- **Question:**
  What is the difference between overriding Equals and == for object comparison in C# struct and class ?

- **Question:**
  Why is overriding ToString beneficial when working with custom classes?

- **Problem:**
  Write a generic method Max in the Helper class that takes two arguments and returns the greater value. Demonstrate the usage of this method with integers, doubles, and strings.
- **Question:**
  Can generics be constrained to specific types in C#? Provide an example.

- **Problem:**
  Add a new method ReplaceArray in the Helper2<T> class that replaces all occurrences of a specified value in an array with another value. Demonstrate with both integer and string arrays.
- **Question:**
  What are the key differences between generic methods and generic classes?

- **Problem:**
  Write a non-generic Swap method for a custom struct Rectangle with properties Length and Width. Create instances of Rectangle and demonstrate swapping their values.
- **Question:**
  Why might using a generic swap method be preferable to implementing custom methods for each type?

- **Problem:**
  Create a Department class and use it to add a Department property to the Employee class. Demonstrate searching an array of employees by department using the SearchArray method.
- **Question:**
  How can overriding Equals for the Department class improve the accuracy of searches?

- **Problem:**
  Create a custom struct Circle with properties Radius and Color. Compare its instances using both == and Equals. Demonstrate the difference in behavior when the same operations are performed on instances of a Circle class.
- **Question:**
  Why is == not implemented by default for structs?

Part02

1- LinkedIn article about class types ?
2- What we mean by Generalization concept using Generics ?
3- What we mean by hierarchy design in real business ?

Problem 1: Generic Method for Reversing an Array

Description: Create a generic method to reverse the elements of an array.
Requirements:

- The method should accept an array of any type and return a new array with the elements in reverse order.
- Ensure the method works for different types such as integers, strings, and custom objects.

Problem 2: Generic Class for a Stack

Description: Implement a generic class for a stack data structure.
Requirements:

- The class should support standard stack operations such as push, pop, and peek.
- Ensure type safety using generics.

Problem 3: Generic Method for Swapping Elements

Description: Implement a generic method to swap two elements in an array.
Requirements:

- The method should accept an array and two indices.
- Swap the elements at the given indices.

Problem 4: Generic Method for Finding Maximum Element

Description: Implement a generic method to find the maximum element in an array.
Requirements:

- The method should accept an array of any type that implements IComparable.
- Return the maximum element in the array.

Part03 Bonus

1- self study report

2- what is Event deriven programming