Kursusgang 3 - Interfaces

1 Sorting cars

```
namespace KG3;
class Car
{
    public string Make { get; private set; }
    public string Model { get; private set; }
    public decimal Price { get; private set; }

    public Car(string make, string model, decimal price)
    {
        Make = make;
        Model = model;
        Price = price;
    }
}
```

Listing 1: Car class

```
using KG3;

List<Car> cars = new List<Car>()
{
    new Car("Skoda", "Fabia", 50000m),
    new Car("Skoda", "Octavia", 60000m),
    new Car("Fiat", "500", 12345m),
    new Car("Ford", "Mustang", 9000000m),
    new Car("Ford", "Mustang", 9000001m)
};
cars.Sort();
Console.WriteLine("Sorted by price");
foreach (Car car in cars)
{
    Console.WriteLine($" {car.Make} {car.Model} {car.Price}");
}
```

Listing 2: The main program for cars. A list of cars must be sorted.

(a) Consider the given class Car in Listing 1. Implement the IComparable interface on Car, to sort the list of cars by Price. The list of cars can be seen in Listing 2.

(b) Make a class CarComparer which implements the IComparer<Car> interface to sort cars by Make, Model, and, lastly, by Price.

(c) Change the CarComparer from the previous exercise such that it sorts by Make, Model, and Price in that order; however, reverse the order for which objects are sorted by Price.

2 Implementing tax through interfaces

(a) Consider the classes found here. Program an interface ITaxable with a readonly property TaxValue. Make sure the classes House and Bus implements this ITaxable interface.

(b) Demonstrate that taxable house objects and taxable bus objects can be used together as objects of type ITaxable.

3 GameObject program

(a) Consider the code found here. Restructure the GameObject program such that class Die and class Card both inherit an abstract class GameObject. You should write the abstarct class GameObject.