Const

```
public enum VehicleType
{
    Personal = 1, Family, Sports, Racing
}
```

Vehicle

```
public abstract class Vehicle
{
    protected Vehicle()
    {

    }
    protected Vehicle(string modelno, int yearmake, int
enginecapacityincc, int noofGear, VehicleType vehicleType)
    {
        this.ModelNo = modelno;
        this.YearMake = yearmake;
        this.EngineCapacityInCC = enginecapacityincc;
        this.NoOfGear = noofGear;
        this.VehicleType = vehicleType;
    }
    public string ModelNo { get; set; }
    public int YearMake { get; set; }
    public int EngineCapacityInCC { get; set; }
    public int NoOfGear { get; set; }
    public VehicleType VehicleType { get; set; }

    public abstract string Details();

}
```

Twowheeler

```
namespace Generic_Approach
{
    public class TwoWheeler : Vehicle, iexteriordesign
    {
        private readonly List<string> exDesign = new
List<string>();
        public TwoWheeler()
```

```
        {

        }
        public TwoWheeler(string modelno, int yearmake, int
enginecapacityincc, int noofGear, VehicleType vehicleType, double
mileage, string startingMethod) :
            base(modelno, yearmake, enginecapacityincc, noofGear,
vehicleType)
        {
            this.Mileage = mileage;
            this.StartingMethod = startingMethod;
        }
        public double Mileage { get; set; }
        public string StartingMethod { get; set; }

        public void AddExteriorDesign(params string[] designs)
        {
            this.exDesign.AddRange(designs);
        }

        public void AddInteriorDesign(params string[] design)
        {
            throw new NotImplementedException();
        }

        public override string Details()
        {
            return
$"{ModelNo},{YearMake},{EngineCapacityInCC},{NoOfGear}\n{Mileage}
,{StartingMethod}";
        }

        public string GetExteriorDesign()
        {
            return string.Join(", ", exDesign);
        }

        public string GetInteriorDesign()
        {
            throw new NotImplementedException();
        }
    }
}
```

Fourwheeler

```
namespace Generic_Approach
{
    public class FourWheeler : Vehicle, iinteriordesign
    {
        private readonly List<string> inDesign = new
List<string>();
        public FourWheeler()
        {

        }
        public FourWheeler(string modelno, int yearmake, int
enginecapacityincc, int noofGear, VehicleType vehicleType, int
noofSeats) :
            base(modelno, yearmake, enginecapacityincc, noofGear,
vehicleType)
        {
            this.NoOfSeats = noofSeats;
        }
        public int NoOfSeats { get; set; }
        public void AddInteriorDesign(params string[] design)
        {
            this.inDesign.AddRange(design);
        }

        public override string Details()
        {
            return
$"{ModelNo},{YearMake},{EngineCapacityInCC},{NoOfGear}\n{NoOfSeat
s}";
        }

        public string GetInteriorDesign()
        {
            return string.Join(", ", inDesign);
        }
    }
}
```

Iixteriordesign

```
namespace Generic_Approach
{
    internal interface iexteriordesign
    {
        void AddInteriorDesign(params string[] design);
        string GetInteriorDesign();
    }
}
```

Iinteriordesign

```
namespace Generic_Approach
{
    public interface iinteriordesign
    {
        void AddInteriorDesign(params string[] design);
        string GetInteriorDesign();
    }
}
```

Car

```
namespace Generic_Approach
{
    public class Car : FourWheeler
    {
        public Car()
        {

        }
        public Car(string modelno, int yearmake, int
enginecapacityincc, int noofGear, VehicleType vehicleType, int
noofSeats, int noofDoor) :
            base(modelno, yearmake, enginecapacityincc, noofGear,
vehicleType, noofDoor)
        {
            this.NoOfDoor = noofDoor;
        }
        public int NoOfDoor { get; set; }
```

```
        public override string Details()
        {
            return $"{base.Details()}\n noof door{NoOfDoor}";
        }
    }
}
```

Motorcycle

```
namespace Generic_Approach
{
    public class MotorCycle : TwoWheeler
    {
        public MotorCycle()
        {

        }
        public MotorCycle(string modelno, int yearmake, int
enginecapacityincc, int noofGear, VehicleType vehicleType, double
mileage, string startingMethod,
            int maxpower, int maxtorque, string coling, string
frontbrake, string rearbrake) :
            base(modelno, yearmake, enginecapacityincc, noofGear,
vehicleType, mileage, startingMethod)
        {
            this.MaxPower = maxpower;
            this.MaxTorque = maxtorque;
            this.Coling = coling;
            this.FrontBrake = frontbrake;
            this.RearBrake = rearbrake;
        }
        public int MaxPower { get; set; }
        public int MaxTorque { get; set; }
        public string Coling { get; set; }
        public string FrontBrake { get; set; }
        public string RearBrake { get; set; }
        public override string Details()
        {
            return $"{base.Details()}\n max power{MaxPower}, max
torque{MaxTorque}, coling{Coling}, front brake{FrontBrake}, rear
brake{RearBrake}";
        }
```

```
    }
}
```

IGenericDetail

```
namespace Generic_Approach
{
    public interface IGenericDetail<T>
    {
        string GetDetails<T1>(T1 obj);
    }
}
```

GenericDetailImpl

```
namespace Generic_Approach
{
    public class GenericDetailImpl<T> : IGenericDetail<T>
    {
        public string GetDetails<T1>(T1 obj)
        {
            if (obj is Vehicle)
            {
                Vehicle v = obj as Vehicle;
                return v.Details();
            }
            else
            {
                return "Not a Vehicle!!";
            }
        }
    }
}
```

Program

```
namespace Generic_Approach
{
```

```
    internal class Program
    {
        static void Main(string[] args)
        {

            Console.WriteLine();

Console.WriteLine("****************************************Motor
Cycle****************************************");
            Console.WriteLine();

            MotorCycle m = new MotorCycle("R15", 2002, 180, 6,
VehicleType.Personal, 30.00, "Self+Kick", 100, 6, "Light",
"Disc", "Drum");
            m.AddExteriorDesign("Vip-Horn", "Fog_Light");
            GenericDetailImpl<MotorCycle> gd = new
GenericDetailImpl<MotorCycle>();
            Console.WriteLine(gd.GetDetails<MotorCycle>(m));
            Console.WriteLine(m.GetExteriorDesign());
            Console.WriteLine();
            Console.ReadKey();
        }
    }
}
```

LINQ

Product

```
namespace Linq
{
    public class Product
    {
        public int ProductID { get; set; }
        public string Name { get; set; }
        public string ProductNumber { get; set; }
        public string Color { get; set; }
        public double StandardCost { get; set; }
        public double ListPrice { get; set; }
        public int Size { get; set; }
```

```
        public double Weight { get; set; }
        public int ProductCategoryID { get; set; }
        public int ProductModelID { get; set; }
    }
}
```

ProductCategory

```
namespace Linq
{
    public class ProductCategory
    {
        public int ProductCategoryID { get; set; }
        public string Name { get; set; }
    }
}
```

ProductModel

```
namespace Linq
{
    public class ProductModel
    {
        public int ProductModelID { get; set; }
        public string Name { get; set; }
    }
}
```

Program

```
namespace Linq
{
    internal class Program
    {
        static void Main(string[] args)
        {
```

```
            List<ProductCategory> categories = new
List<ProductCategory>
            {
                new ProductCategory{ProductCategoryID = 1, Name =
"Bikes"   },
                new ProductCategory{ProductCategoryID = 2, Name =
"Components"  },
                new ProductCategory{ProductCategoryID = 3, Name =
"Clothing"    },
                new ProductCategory{ProductCategoryID = 4, Name =
"Accessories" },
                new ProductCategory{ProductCategoryID = 5, Name =
"Mountain Bikes"  },
                new ProductCategory{ProductCategoryID = 6, Name =
"Road Bikes"  },
                new ProductCategory{ProductCategoryID = 7, Name =
"Touring Bikes"   },
                new ProductCategory{ProductCategoryID = 8, Name =
"Handlebars"  },
                new ProductCategory{ProductCategoryID = 9, Name =
"Bottom Brackets" },
                new ProductCategory{ProductCategoryID = 10, Name
= "Brakes" },
                new ProductCategory{ProductCategoryID = 11, Name
= "Chains" },
                new ProductCategory{ProductCategoryID = 12, Name
= "Cranksets"  },
                new ProductCategory{ProductCategoryID = 13, Name
= "Derailleurs"    },
                new ProductCategory{ProductCategoryID = 14, Name
= "Forks"  },
                new ProductCategory{ProductCategoryID = 15, Name
= "Headsets"   },
                new ProductCategory{ProductCategoryID = 16, Name
= "Mountain Frames"    },
                new ProductCategory{ProductCategoryID = 17, Name
= "Pedals" },
                new ProductCategory{ProductCategoryID = 18, Name
= "Road Frames"    },
                new ProductCategory{ProductCategoryID = 19, Name
= "Saddles"    },
                new ProductCategory{ProductCategoryID = 20, Name
= "Touring Frames" },
```

```
                new ProductCategory{ProductCategoryID = 21, Name
= "Wheels" },
                new ProductCategory{ProductCategoryID = 22, Name
= "Bib-Shorts" },
                new ProductCategory{ProductCategoryID = 23, Name
= "Caps"   },
                new ProductCategory{ProductCategoryID = 24, Name
= "Gloves" },
                new ProductCategory{ProductCategoryID = 25, Name
= "Jerseys"    },
                new ProductCategory{ProductCategoryID = 26, Name
= "Shorts" },
                new ProductCategory{ProductCategoryID = 27, Name
= "Socks"  },

            };

            List<ProductModel> models = new List<ProductModel>
            {
                new ProductModel{ProductModelID = 1, Name =
"Classic Vest"   },
                new ProductModel{ProductModelID = 2, Name =
"Cycling Cap"    },
                new ProductModel{ProductModelID = 3, Name =
"Full-Finger Gloves" },
                new ProductModel{ProductModelID = 4, Name =
"Half-Finger Gloves" },
                new ProductModel{ProductModelID = 5, Name = "HL
Mountain Frame"  },
                new ProductModel{ProductModelID = 6, Name = "HL
Road Frame"  },
                new ProductModel{ProductModelID = 7, Name = "HL
Touring Frame"   },
                new ProductModel{ProductModelID = 8, Name = "LL
Mountain Frame"  },
                new ProductModel{ProductModelID = 9, Name = "LL
Road Frame"  },
                new ProductModel{ProductModelID = 10, Name = "LL
Touring Frame"  },
                new ProductModel{ProductModelID = 11, Name =
"Long-Sleeve Logo Jersey"   },
                new ProductModel{ProductModelID = 12, Name =
"Men's Bib-Shorts"  },
```

```
                new ProductModel{ProductModelID = 13, Name =
"Men's Sports Shorts"   },
                new ProductModel{ProductModelID = 14, Name = "ML
Mountain Frame" },
                new ProductModel{ProductModelID = 15, Name = "ML
Mountain Frame-W"   },

            };

            List<Product> products = new List<Product>
            {
                new Product{ProductID = 2, Name = "LL Road Frame
- Red, 48", ProductNumber = "FR-R38R-48", Color = "Red",
StandardCost = 187.1571, ListPrice = 337.22, Size = 48, Weight =
1070.47, ProductCategoryID = 18, ProductModelID = 9  },
                new Product{ProductID = 3, Name = "LL Road Frame
- Red, 52", ProductNumber = "FR-R38R-52", Color = "Red",
StandardCost = 187.1571, ListPrice = 337.22, Size = 52, Weight =
1088.62, ProductCategoryID = 18, ProductModelID = 9  },
                new Product{ProductID = 1, Name = "LL Road Frame
- Red, 44", ProductNumber = "FR-R38R-44", Color = "Red",
StandardCost = 187.1571, ListPrice = 337.22, Size = 44, Weight =
1052.33, ProductCategoryID = 18, ProductModelID = 9  },
                new Product{ProductID = 4, Name = "LL Road Frame
- Red, 58", ProductNumber = "FR-R38R-58", Color = "Red",
StandardCost = 187.1571, ListPrice = 337.22, Size = 58, Weight =
1115.83, ProductCategoryID = 18, ProductModelID = 9  },
                new Product{ProductID = 14, Name = "HL Mountain
Frame - Black, 44", ProductNumber = "FR-M94B-44", Color =
"Black", StandardCost = 699.0928, ListPrice = 1349.6, Size = 44,
Weight = 1251.91, ProductCategoryID = 16, ProductModelID = 5 },
                new Product{ProductID = 15, Name = "HL Mountain
Frame - Black, 48", ProductNumber = "FR-M94B-48", Color =
"Black", StandardCost = 699.0928, ListPrice = 1349.6, Size = 48,
Weight = 1270.05, ProductCategoryID = 16, ProductModelID = 5 },

            };


            //Join
            Console.WriteLine();
            Console.WriteLine("Query");
            Console.WriteLine("====================");
            (
```

```
            from p in products
            join c in categories on p.ProductCategoryID equals
c.ProductCategoryID
            join m in models on p.ProductModelID equals
m.ProductModelID
            select new { p.ProductID, p.Name, category = c.Name,
model = m.Name, p.Color, p.Weight, p.Size, p.ListPrice }
            ).ToList()
            .ForEach(p =>
            {
                Console.WriteLine($"{p.ProductID}, {p.Name},
category: {p.category}");
            });

            Console.WriteLine();
            Console.WriteLine("Lambda");
            Console.WriteLine("====================");

            products.Where(p => p.Color == "Black")
                    .ToList()
                    .ForEach(p =>
                    {
                        var category =
categories.FirstOrDefault(x => x.ProductCategoryID ==
p.ProductCategoryID);
                        var model = models.FirstOrDefault(X =>
X.ProductModelID == p.ProductModelID);


Console.WriteLine($"{p.ProductID},{p.Name},Model: {model.Name}");
                    });

            Console.ReadKey();
        }
    }
}
```