Inheritance in object-oriented programming is the ability of a class (child) to get attributes from another class (parent). This allows the child class to use the code in the parent class and extend it or change it. The main benefit of this concept is reusability as the child class will be able to use parent’s class code without rewriting it. This makes the code more simple and easier to maintain as it reduces redundancy and ensures consistency.

An example of an application where inheritance can be applied is a graphical user interface (GUI) framework a parent class (“Car”) can have the main code and its children (“ElectricCar”, “GasCar”) will have specific behaviours.

A code example of the application suggested above, could be:  
  
using System;

**// Base class**

public class Car

{

public string Make { get; set; }

public string Model { get; set; }

public void Drive()

{

Console.WriteLine($"{Make} {Model} is driving.");

}

}

**// Derived class**

public class ElectricCar : Car

{

public int BatteryLevel { get; set; }

public void ChargeBattery()

{

Console.WriteLine($"{Make} {Model} charging battery.");

}

}

**// Another derived class**

public class GasCar : Car

{

public int FuelLevel { get; set; }

public void Refuel()

{

Console.WriteLine($"{Make} {Model} is refuelling.");

}

}

class Program

{

static void Main(string[] args)

{

ElectricCar tesla = new ElectricCar();

tesla.Make = "Tesla";

tesla.Model = "Model S";

tesla.BatteryLevel = 80;

GasCar toyota = new GasCar();

toyota.Make = "Toyota";

toyota.Model = "Camry";

toyota.FuelLevel = 50;

**// Using inherited method**

tesla.Drive();

toyota.Drive();

**// Using methods specific to derived classes**

tesla.ChargeBattery();

toyota.Refuel();

}

}