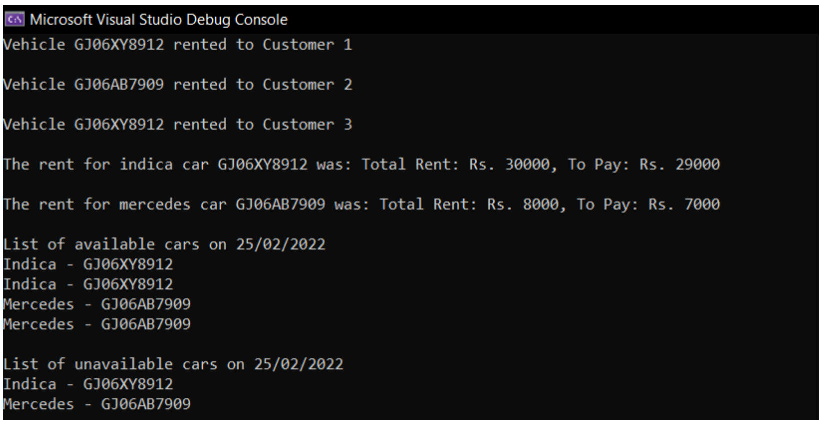
# Practical – 5

## Generic Class

**Code:**

```
namespace Pract4 {

    public class Indica
    {
        public int seater, rentType; // Rent Type: 1 per day, 2 per km
        public double rentPerUnit, age;
        public string number;
        public DateOnly lastMaintenanceDate;
    }
    public class Qualis
    {
        public int seater, rentType;
        public double rentPerUnit, age;
        public string number = string.Empty;
        public DateOnly lastMaintenanceDate;
    }
    public class HarleyDavidson
    {
        public int rentType;
        public double rentPerUnit, age;
        public string number = string.Empty;
        public DateOnly lastMaintenanceDate;
    }

    public class MBenzEclass
    {
        public int seater, rentType;
        public double rentPerUnit, age;
        public string number = string.Empty;
        public DateOnly lastMaintenanceDate;
    }

    public class RentedVehicle<T> where T: new()
    {

        public DateOnly startDateofRent, endDateofRent, maintanenceDate;
        public double noofkmstravelled, advancepayment;
        public string custname = string.Empty;
        public T vehicle;

        public RentedVehicle(){
            vehicle = new T();
        }

        public void giveForRent(string custname, DateOnly startDate, DateOnly
endDate,double advancePayment)
        {

            this.advancepayment = advancePayment;
            this.startDateofRent = startDate;
            this.endDateofRent = endDate;
            this.custname = custname;

        }
        public bool checkVehicleAvailable(DateOnly startDate, DateOnly endDate)
        {
            if (this.startDateofRent != DateOnly.MinValue && this.endDateofRent 
!= DateOnly.MinValue)
            {
                if ((startDate < this.startDateofRent && endDate < 
this.endDateofRent) || (startDate > this.startDateofRent && endDate > 
this.endDateofRent))
                {
                    if (maintanenceDate != DateOnly.MinValue)
                    {
                        if (this.maintanenceDate >= startDate && 
this.maintanenceDate <= endDate)
                        {
                            return false;
                        }
                        else
                        {
                            return true;
                        }
                    }
                    else
                    {
                        return true;
                    }
                }
                else
                {
                    return false;
                }
            }
            else
            {
                return true;
            }
        }

        public string calculateRent(int noOfDays, double kmTravelled, int 
rentType, double ratePerUnit)
        {
            if(rentType == 1)
            {
                return String.Format("Total Rent: Rs. {0}, To Pay: Rs. {1}", 
noOfDays * ratePerUnit, (noOfDays * ratePerUnit) - this.advancepayment);
            }
            else
            {
                return String.Format("Total Rent: Rs. {0}, To Pay: Rs. {1}", 
kmTravelled * ratePerUnit, (kmTravelled * ratePerUnit) - this.advancepayment);
            }
        }
    }

    public class RentCar
    {
        public static void Main()
        {


            List<string> availableCars = new List<string>();
            List<string> unavailableCars = new List<string>();

            RentedVehicle<Indica>[] indicas = new RentedVehicle<Indica>[5];
            indicas[0] = new RentedVehicle<Indica>();
            indicas[0].vehicle.rentPerUnit = 10;
            indicas[0].vehicle.rentType = 2;
            indicas[0].vehicle.seater = 4;
            indicas[0].vehicle.lastMaintenanceDate = 
DateOnly.ParseExact("10/01/2022","dd/MM/yyyy");
            indicas[0].vehicle.age = 2;
            indicas[0].vehicle.number = "GJ06XY8912";
            indicas[0].maintanenceDate = 
indicas[0].vehicle.lastMaintenanceDate.AddMonths(6);

            indicas[1] = new RentedVehicle<Indica>();
            indicas[1].vehicle.rentPerUnit = 9;
            indicas[1].vehicle.rentType = 2;
            indicas[1].vehicle.seater = 4;
            indicas[1].vehicle.lastMaintenanceDate = 
DateOnly.ParseExact("24/01/2022","dd/MM/yyyy");
            indicas[1].vehicle.age = 3;
            indicas[1].vehicle.number = "GJ06YY9901";
            indicas[1].maintanenceDate = 
indicas[1].vehicle.lastMaintenanceDate.AddMonths(6);

            indicas[2] = new RentedVehicle<Indica>();
            indicas[2].vehicle.rentPerUnit = 11;
            indicas[2].vehicle.rentType = 2;
            indicas[2].vehicle.seater = 4;
            indicas[2].vehicle.lastMaintenanceDate = 
DateOnly.ParseExact("01/02/2022","dd/MM/yyyy");
            indicas[2].vehicle.age = 1;
            indicas[2].vehicle.number = "GJ06YY4982";
            indicas[2].maintanenceDate = 
indicas[2].vehicle.lastMaintenanceDate.AddMonths(6);

            RentedVehicle<MBenzEclass>[] mercedes = new 
RentedVehicle<MBenzEclass>[5];

            mercedes[0] = new RentedVehicle<MBenzEclass>();
            mercedes[0].vehicle.rentPerUnit = 2000;
            mercedes[0].vehicle.rentType = 1;
            mercedes[0].vehicle.seater = 5;
            mercedes[0].vehicle.lastMaintenanceDate = 
DateOnly.ParseExact("30/12/2021","dd/MM/yyyy");
            mercedes[0].vehicle.age = 1;
            mercedes[0].vehicle.number = "GJ06AB7909";
            mercedes[0].maintanenceDate = 
mercedes[0].vehicle.lastMaintenanceDate.AddMonths(6);

            mercedes[1] = new RentedVehicle<MBenzEclass>();
            mercedes[1].vehicle.rentPerUnit = 2000;
            mercedes[1].vehicle.rentType = 1;
            mercedes[1].vehicle.seater = 5;
            mercedes[1].vehicle.lastMaintenanceDate = 
DateOnly.ParseExact("15/12/2021","dd/MM/yyyy");
            mercedes[1].vehicle.age = 1;
            mercedes[1].vehicle.number = "GJ06AB1234";
            mercedes[1].maintanenceDate = 
mercedes[1].vehicle.lastMaintenanceDate.AddMonths(6);

            mercedes[2] = new RentedVehicle<MBenzEclass>();
            mercedes[2].vehicle.rentPerUnit = 2500;
            mercedes[2].vehicle.rentType = 1;
            mercedes[2].vehicle.seater = 5;
            mercedes[2].vehicle.age = 0.5;
            mercedes[2].vehicle.number = "GJ06AB0021";
            mercedes[2].maintanenceDate = 
mercedes[2].vehicle.lastMaintenanceDate.AddMonths(6);

            RentedVehicle<Qualis>[] qualis = new RentedVehicle<Qualis>[5];
            RentedVehicle<HarleyDavidson>[] harleys = new 
RentedVehicle<HarleyDavidson>[5];


            // Giving indica on rent
            bool vehicleRented = false;
            for(int i =0 ; i < 3; i++)
            {
                
if(indicas[i].checkVehicleAvailable(DateOnly.ParseExact("20/02/2022","dd/MM/yyyy"
), DateOnly.ParseExact("25/02/2022", "dd/MM/yyyy"))){
                    indicas[i].giveForRent("Customer 1", 
DateOnly.ParseExact("20/02/2022", "dd/MM/yyyy"), 
DateOnly.ParseExact("25/02/2022", "dd/MM/yyyy"), 1000);
                    vehicleRented = true;
                    Console.WriteLine("Vehicle {0} rented to {1}", 
indicas[i].vehicle.number, indicas[i].custname);
                    break;
                }
            }
            if (!vehicleRented)
            {
                Console.WriteLine("No vehicle available");
            }
            vehicleRented = false;

            // Giving mercedes on rent

            for (int i = 0; i < 3; i++)
            {
                if 
(mercedes[i].checkVehicleAvailable(DateOnly.ParseExact("20/02/2022", 
"dd/MM/yyyy"), DateOnly.ParseExact("25/02/2022", "dd/MM/yyyy"))){
                    mercedes[i].giveForRent("Customer 2", 
DateOnly.ParseExact("20/02/2022", "dd/MM/yyyy"), 
DateOnly.ParseExact("25/02/2022", "dd/MM/yyyy"), 1000);
                    vehicleRented = true;
                    Console.WriteLine("\nVehicle {0} rented to {1}", 
mercedes[i].vehicle.number, mercedes[i].custname);
                    break;
                }
            }
            if (!vehicleRented)
            {
                Console.WriteLine("\nNo vehicle available");
            }
            vehicleRented=false;

            // Giving another indica on rent

            for (int i = 0; i < 3; i++)
            {
                if 
(indicas[i].checkVehicleAvailable(DateOnly.ParseExact("23/02/2022", 
"dd/MM/yyyy"), DateOnly.ParseExact("28/02/2022", "dd/MM/yyyy")))
                {
                    indicas[i].giveForRent("Customer 3", 
DateOnly.ParseExact("23/02/2022", "dd/MM/yyyy"), 
DateOnly.ParseExact("28/02/2022", "dd/MM/yyyy"), 1000);
                    vehicleRented = true;
                    Console.WriteLine("\nVehicle {0} rented to {1}", 
indicas[i].vehicle.number, indicas[i].custname);
                    break;
                }
            }
            if (!vehicleRented)
            {
                Console.WriteLine("No vehicle available");
            }
            vehicleRented = false;

            // Calculate rent for indica 1


            Console.WriteLine("\nThe rent for indica car {0} was:
{1}",indicas[0].vehicle.number,indicas[0].calculateRent(4, 3000,
indicas[0].vehicle.rentType, indicas[0].vehicle.rentPerUnit));

            // Calculate rent for mercedes

            Console.WriteLine("\nThe rent for mercedes car {0} was: {1}",
mercedes[0].vehicle.number, mercedes[0].calculateRent(4, 3000,
mercedes[0].vehicle.rentType, mercedes[0].vehicle.rentPerUnit));


            availableCars.Clear();
            unavailableCars.Clear();
            for (int i = 0; i < 3; i++)
            {
                if
(indicas[i].checkVehicleAvailable(DateOnly.ParseExact("25/02/2022",
"dd/MM/yyyy"), DateOnly.ParseExact("25/02/2022", "dd/MM/yyyy")))
                {
                    availableCars.Add("Indica - " + indicas[0].vehicle.number);
                }
                else
                {
                    unavailableCars.Add("Indica - " + indicas[0].vehicle.number);
                }
            }


            for (int i = 0; i < 3; i++)
            {
                if
(mercedes[i].checkVehicleAvailable(DateOnly.ParseExact("25/02/2022",
"dd/MM/yyyy"), DateOnly.ParseExact("25/02/2022", "dd/MM/yyyy")))
                {
                    availableCars.Add("Mercedes - " +
mercedes[0].vehicle.number);
                }
                else
                {
                    unavailableCars.Add("Mercedes - " +
mercedes[0].vehicle.number);
                }
            }

            Console.WriteLine("\nList of available cars on 25/02/2022");
            foreach(string t in availableCars)
            {
                Console.WriteLine(t);
            }

            Console.WriteLine("\nList of unavailable cars on 25/02/2022");
            foreach (string t in unavailableCars)
            {
                Console.WriteLine(t);
            }
            }
      }
}
```

## Output:



```
Vehicle GJ06XY8912 rented to Customer 1

Vehicle GJ06AB7909 rented to Customer 2

Vehicle GJ06XY8912 rented to Customer 3

The rent for indica car GJ06XY8912 was: Total Rent: Rs. 30000, To Pay: Rs. 29000

The rent for mercedes car GJ06AB7909 was: Total Rent: Rs. 8000, To Pay: Rs. 7000

List of available cars on 25/02/2022
Indica - GJ06XY8912
Indica - GJ06XY8912
Mercedes - GJ06AB7909
Mercedes - GJ06AB7909

List of unavailable cars on 25/02/2022
Indica - GJ06XY8912
Mercedes - GJ06AB7909
```