Assignment 5:

Create classes with fields as given below:

Indica- type, seater, renttype, rentperunit, number, lastmaintenancedate,

assignment also

age,

Qualis type, seater, renttype, rentperunit, age, number, lastmaintenancedate,

Davidharley renttype, rentperunit, age, number, lastmaintenancedate,

MBenzEclass seater, renttype, rentperunit, age, number, lastmaintenancedate,

Create a class RentedVehicle<T> where T we will use classes

like Indica, Qualis, David Harley, Mercedes Benz E class

RentedVehicle<T> has fields startdateofrent, enddateofrent, noofkestravelled, advancepayment

Given below are some methods that RentedVehicle<T> will implement

GiveForRent-noofkms, startdate, enddate, advancepayment

- CalculateRent

noofkms, noofdays,

. CalculateTotalRent

GetCheckListREntedandAvailableVehicle

GetCheckListvehilcesinmaintainence

ShowAvailability forbookingforgivendate

Sample Code for driver class main method:

RentedVehicle Indica> indical-newRentedVehicle Indica>();

RentedVehicle<Qualis>[] ArrQualis = new RentedVehilce Qualis>[4];

Code:

asdfusing System;

using System.Collections.Generic;

class Programme

{

public class Indica

{

private string type = "4-Wheeler";

private int seater = 5;

private string rent\_type;

private double rent\_per\_unit;

private int age;

private string number;

private DateTime last\_maintanance;

private bool rented = false;

private DateTime today = DateTime.Now;

private int diff\_main;

private int main\_year;

private int main\_month;

private int main\_day;

public Indica(string temp)

{

}

public Indica()

{

Console.WriteLine($"Enter Rent Per Kilo-meter :");

rent\_per\_unit = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter NUmber : ");

number = Console.ReadLine();

Console.WriteLine($"Enter last date for maintainance : ");

main\_year = Convert.ToInt32(Console.ReadLine());

main\_month = Convert.ToInt32(Console.ReadLine());

main\_day = Convert.ToInt32(Console.ReadLine());

last\_maintanance = new DateTime(main\_year, main\_month, main\_day);

Console.WriteLine($"Enter age of Vehicle : ");

age = Convert.ToInt32(Console.ReadLine());

}

public string Type

{

get => type;

set => type = value;

}

public int Seater

{

get => seater;

set => seater = value;

}

public string RentType

{

get => rent\_type;

set => rent\_type = value;

}

public double RentPerUnit

{

get => rent\_per\_unit;

set => rent\_per\_unit = value;

}

public int Age

{

get => age;

set => age = value;

}

public string Number

{

get => number;

set => number = value;

}

public DateTime Last\_maintanance

{

get => last\_maintanance;

set => last\_maintanance = value;

}

public bool Rented

{

get => rented;

set => rented = value;

}

public int DiffMain

{

get => diff\_main;

set => diff\_main = value;

}

}

public class Qualis

{

private string type = "4-Wheeler";

private int seater = 10;

private string rent\_type;

private double rent\_per\_unit;

private int age;

private string number;

private DateTime last\_maintanance;

private bool rented = false;

private DateTime today = DateTime.Now;

private int diff\_main;

private int main\_year;

private int main\_month;

private int main\_day;

public Qualis()

{

Console.WriteLine($"Enter Rent Per Kilo-meter :");

rent\_per\_unit = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter NUmber : ");

number = Console.ReadLine();

Console.WriteLine($"Enter last date for maintainance : ");

main\_year = Convert.ToInt32(Console.ReadLine());

main\_month = Convert.ToInt32(Console.ReadLine());

main\_day = Convert.ToInt32(Console.ReadLine());

last\_maintanance = new DateTime(main\_year, main\_month, main\_day);

Console.WriteLine($"Enter age of Vehicle : ");

age = Convert.ToInt32(Console.ReadLine());

}

public string Type

{

get => type;

set => type = value;

}

public int Seater

{

get => seater;

set => seater = value;

}

public string RentType

{

get => rent\_type;

set => rent\_type = value;

}

public double RentPerUnit

{

get => rent\_per\_unit;

set => rent\_per\_unit = value;

}

public int Age

{

get => age;

set => age = value;

}

public string Number

{

get => number;

set => number = value;

}

public DateTime Last\_maintanance

{

get => last\_maintanance;

set => last\_maintanance = value;

}

public bool Rented

{

get => rented;

set => rented = value;

}

public int DiffMain

{

get => diff\_main;

set => diff\_main = value;

}

}

public class Davidharley

{

private string type = "2-Wheeler";

private int seater = 2;

private string rent\_type;

private double rent\_per\_unit;

private int age;

private string number;

private DateTime last\_maintanance;

private bool rented = false;

private DateTime today = DateTime.Now;

private int diff\_main;

private int main\_year;

private int main\_month;

private int main\_day;

public Davidharley()

{

Console.WriteLine($"Enter Rent Per Kilo-meter :");

rent\_per\_unit = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter NUmber : ");

number = Console.ReadLine();

Console.WriteLine($"Enter last date for maintainance : ");

main\_year = Convert.ToInt32(Console.ReadLine());

main\_month = Convert.ToInt32(Console.ReadLine());

main\_day = Convert.ToInt32(Console.ReadLine());

last\_maintanance = new DateTime(main\_year, main\_month, main\_day);

Console.WriteLine($"Enter age of Vehicle : ");

age = Convert.ToInt32(Console.ReadLine());

}

public string Type

{

get => type;

set => type = value;

}

public int Seater

{

get => seater;

set => seater = value;

}

public string RentType

{

get => rent\_type;

set => rent\_type = value;

}

public double RentPerUnit

{

get => rent\_per\_unit;

set => rent\_per\_unit = value;

}

public int Age

{

get => age;

set => age = value;

}

public string Number

{

get => number;

set => number = value;

}

public DateTime Last\_maintanance

{

get => last\_maintanance;

set => last\_maintanance = value;

}

public bool Rented

{

get => rented;

set => rented = value;

}

public int DiffMain

{

get => diff\_main;

set => diff\_main = value;

}

}

public class Mbenzeclass

{

private string type = "4-Wheeler";

private int seater = 5;

private string rent\_type;

private double rent\_per\_unit;

private int age;

private string number;

private DateTime last\_maintanance;

private bool rented = false;

private DateTime today = DateTime.Now;

private int diff\_main;

private int main\_year;

private int main\_month;

private int main\_day;

public Mbenzeclass()

{

Console.WriteLine($"Enter Rent Per Kilo-meter :");

rent\_per\_unit = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter NUmber : ");

number = Console.ReadLine();

Console.WriteLine($"Enter last date for maintainance : ");

main\_year = Convert.ToInt32(Console.ReadLine());

main\_month = Convert.ToInt32(Console.ReadLine());

main\_day = Convert.ToInt32(Console.ReadLine());

last\_maintanance = new DateTime(main\_year, main\_month, main\_day);

Console.WriteLine($"Enter age of Vehicle : ");

age = Convert.ToInt32(Console.ReadLine());

}

public string Type

{

get => type;

set => type = value;

}

public int Seater

{

get => seater;

set => seater = value;

}

public string RentType

{

get => rent\_type;

set => rent\_type = value;

}

public double RentPerUnit

{

get => rent\_per\_unit;

set => rent\_per\_unit = value;

}

public int Age

{

get => age;

set => age = value;

}

public string Number

{

get => number;

set => number = value;

}

public DateTime Last\_maintanance

{

get => last\_maintanance;

set => last\_maintanance = value;

}

public bool Rented

{

get => rented;

set => rented = value;

}

public int DiffMain

{

get => diff\_main;

set => diff\_main = value;

}

}

public class RentedVehicle<T>

{

DateTime start\_date\_of\_rent;

DateTime end\_date\_of\_rent;

DateTime given\_date;

int no\_of\_kms\_travelled;

double advance\_payment;

TimeSpan no\_days;

int year\_start;

int year\_end;

int month\_start;

int month\_end;

int day\_start;

int day\_end;

int given\_year;

int given\_month;

int given\_day;

public object curr\_obj;

string v\_name;

DateTime today\_date;

public double curr\_rent;

public List<RentedVehicle<Indica>> list = new List<RentedVehicle<Indica>>();

public List<RentedVehicle<Qualis>> q\_list = new List<RentedVehicle<Qualis>>();

public List<RentedVehicle<Davidharley>> d\_list = new List<RentedVehicle<Davidharley>>();

public List<RentedVehicle<Mbenzeclass>> m\_list = new List<RentedVehicle<Mbenzeclass>>();

public RentedVehicle(T vehicle\_obj, string vehicle)

{

v\_name = vehicle;

curr\_obj = (object)vehicle\_obj;

}

public void GiveForRent()

{

Console.WriteLine("Enter Start Date:");

year\_start = Convert.ToInt32(Console.ReadLine());

month\_start = Convert.ToInt32(Console.ReadLine());

day\_start = Convert.ToInt32(Console.ReadLine());

start\_date\_of\_rent = new DateTime(year\_start, month\_start, day\_start);

Console.WriteLine("Enter End Date:");

year\_end = Convert.ToInt32(Console.ReadLine());

month\_end = Convert.ToInt32(Console.ReadLine());

day\_end = Convert.ToInt32(Console.ReadLine());

end\_date\_of\_rent = new DateTime(year\_end, month\_end, day\_end);

Console.WriteLine("Enter No. Of Kms:");

no\_of\_kms\_travelled = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter Advance Payment:");

advance\_payment = Convert.ToDouble(Console.ReadLine());

if (v\_name == "i")

{

((Indica)(this.curr\_obj)).Rented = true;

Console.WriteLine($"{((Indica)(this.curr\_obj)).Number} Inidica Rented Successfully ! ");

}

else if (v\_name == "q")

{

((Qualis)(this.curr\_obj)).Rented = true;

Console.WriteLine($"{((Qualis)(this.curr\_obj)).Number} Qualis Rented Successfully ! ");

}

else if (v\_name == "m")

{

((Mbenzeclass)(this.curr\_obj)).Rented = true;

Console.WriteLine($"{((Mbenzeclass)(this.curr\_obj)).Number} MBenz Rented Successfully ! ");

}

else if (v\_name == "dh")

{

((Davidharley)(this.curr\_obj)).Rented = true;

Console.WriteLine($"{((Davidharley)(this.curr\_obj)).Number} DavidHarley Rented Successfully ! ");

}

}

public void Calculate\_Rent()

{

no\_days = end\_date\_of\_rent.Subtract(start\_date\_of\_rent);

Console.WriteLine($"Total No. Of Days : {no\_days.TotalDays}");

if (v\_name == "i")

{

Console.WriteLine($"Total Rent : {(no\_of\_kms\_travelled \* ((Indica)(this.curr\_obj)).RentPerUnit) +

(no\_days.TotalDays \* 10)}");

Console.WriteLine($"Rent Remaining : {(no\_of\_kms\_travelled \* ((Indica)(this.curr\_obj)).RentPerUnit)

+ (no\_days.TotalDays \* 10) - advance\_payment}");

}

else if (v\_name == "q")

{

Console.WriteLine($"Total Rent : {(no\_of\_kms\_travelled \* ((Qualis)(this.curr\_obj)).RentPerUnit) +

(no\_days.TotalDays \* 10)}");

Console.WriteLine($"Rent Remaining : {(no\_of\_kms\_travelled \* ((Qualis)(this.curr\_obj)).RentPerUnit)

+ (no\_days.TotalDays \* 10) - advance\_payment}");

}

else if (v\_name == "m")

{

Console.WriteLine($"Total Rent : {(no\_of\_kms\_travelled \* ((Mbenzeclass)(this.curr\_obj)).RentPerUnit)

+ (no\_days.TotalDays \* 10)}");

Console.WriteLine($"Rent Remaining : {(no\_of\_kms\_travelled \*

((Mbenzeclass)(this.curr\_obj)).RentPerUnit) + (no\_days.TotalDays \* 10) - advance\_payment}");

}

else if (v\_name == "dh")

{

Console.WriteLine($"Total Rent : {(no\_of\_kms\_travelled \* ((Davidharley)(this.curr\_obj)).RentPerUnit)

+ (no\_days.TotalDays \* 10)}");

Console.WriteLine($"Rent Remaining : {(no\_of\_kms\_travelled \*

((Davidharley)(this.curr\_obj)).RentPerUnit) + (no\_days.TotalDays \* 10) - advance\_payment}");

}

}

public void CalculateTotalRentToday()

{

today\_date = DateTime.Now;

if (DateTime.Compare(today\_date, end\_date\_of\_rent) <= 0)

{

if (v\_name == "i")

{

curr\_rent = (today\_date.Subtract(start\_date\_of\_rent).TotalDays) \*

((Indica)(this.curr\_obj)).RentPerUnit;

}

else if (v\_name == "q")

{

curr\_rent = (today\_date.Subtract(start\_date\_of\_rent).TotalDays) \*

((Qualis)(this.curr\_obj)).RentPerUnit;

}

else if (v\_name == "m")

{

curr\_rent = (today\_date.Subtract(start\_date\_of\_rent).TotalDays) \*

((Mbenzeclass)(this.curr\_obj)).RentPerUnit;

}

else if (v\_name == "dh")

{

curr\_rent = (today\_date.Subtract(start\_date\_of\_rent).TotalDays) \*

((Davidharley)(this.curr\_obj)).RentPerUnit;

}

Console.WriteLine($"Rent At Today : {curr\_rent}");

}

else

{

Console.WriteLine($"Rent At Today : - ");

}

}

public void GetCheckListRentedAndAvailableVehicle()

{

if (v\_name == "i")

{

if (((Indica)(this.curr\_obj)).Rented)

{

Console.WriteLine($"{((Indica)(this.curr\_obj)).Number} Indica Not Available...");

}

else

{

Console.WriteLine($"{((Indica)(this.curr\_obj)).Number} Indica Available...");

}

}

else if (v\_name == "q")

{

if (((Qualis)(this.curr\_obj)).Rented)

{

Console.WriteLine($"{((Qualis)(this.curr\_obj)).Number} Qualis Not Available...");

}

else

{

Console.WriteLine($"{((Qualis)(this.curr\_obj)).Number} Qualis Available...");

}

}

else if (v\_name == "m")

{

if (((Mbenzeclass)(this.curr\_obj)).Rented)

{

Console.WriteLine($"{((Mbenzeclass)(this.curr\_obj)).Number} MBenz Not Available...");

}

else

{

Console.WriteLine($"{((Mbenzeclass)(this.curr\_obj)).Number} MBenz Available...");

}

}

else if (v\_name == "dh")

{

if (((Davidharley)(this.curr\_obj)).Rented)

{

Console.WriteLine($"{((Davidharley)(this.curr\_obj)).Number} Davidharley Not Available...");

}

else

{

Console.WriteLine($"{((Davidharley)(this.curr\_obj)).Number} Davidharley Available...");

}

}

}

public void GetCheckListVehiclesInmaintanance()

{

if (v\_name == "i")

{

if (DateTime.Compare(DateTime.Today, ((Indica)(this.curr\_obj)).Last\_maintanance) > 0)

{

Console.WriteLine($"{((Indica)(this.curr\_obj)).Number} Indica Under Maintainance");

}

}

else if (v\_name == "q")

{

if (DateTime.Compare(DateTime.Today, ((Qualis)(this.curr\_obj)).Last\_maintanance) > 0)

{

Console.WriteLine($"{((Qualis)(this.curr\_obj)).Number} Qualis Under Maintainance");

}

}

else if (v\_name == "m")

{

if (DateTime.Compare(DateTime.Today, ((Mbenzeclass)(this.curr\_obj)).Last\_maintanance) > 0)

{

Console.WriteLine($"{((Mbenzeclass)(this.curr\_obj)).Number} Mbenz Under Maintainance");

}

}

else if (v\_name == "dh")

{

if (DateTime.Compare(DateTime.Today, ((Davidharley)(this.curr\_obj)).Last\_maintanance) > 0)

{

Console.WriteLine($"{((Davidharley)(this.curr\_obj)).Number} Davidharley Under Maintainance");

}

}

}

public void ShowAvailabilityforbookingforgivendate()

{

Console.WriteLine("Enter Date:");

given\_year = Convert.ToInt32(Console.ReadLine());

given\_month = Convert.ToInt32(Console.ReadLine());

given\_day = Convert.ToInt32(Console.ReadLine());

given\_date = new DateTime(given\_year, given\_month, given\_day);

if (DateTime.Compare(given\_date, end\_date\_of\_rent) > 0)

{

if (v\_name == "i")

{

Console.WriteLine($"{((Indica)(this.curr\_obj)).Number} Indica Available At Given Date...");

}

else if (v\_name == "q")

{

Console.WriteLine($"{((Qualis)(this.curr\_obj)).Number} Qualis Available At Given Date...");

}

else if (v\_name == "m")

{

Console.WriteLine($"{((Mbenzeclass)(this.curr\_obj)).Number} MBenz Available At Given Date...");

}

else if (v\_name == "dh")

{

Console.WriteLine($"{((Davidharley)(this.curr\_obj)).Number} Davidharley Available At Given Date...");

}

}

}

}

static void Main()

{

Indica indica = new Indica("TEMP");

RentedVehicle<Indica> d1 = new RentedVehicle<Indica>(indica, "d");

d1.list.Add(new RentedVehicle<Indica>(new Indica(), "i"));

d1.list.Add(new RentedVehicle<Indica>(new Indica(), "i"));

d1.q\_list.Add(new RentedVehicle<Qualis>(new Qualis(), "q"));

d1.m\_list.Add(new RentedVehicle<Mbenzeclass>(new Mbenzeclass(), "m"));

d1.d\_list.Add(new RentedVehicle<Davidharley>(new Davidharley(), "dh"));

d1.list[0].GiveForRent();

d1.list[0].Calculate\_Rent();

d1.list[0].CalculateTotalRentToday();

Console.WriteLine("\n");

foreach (var ind in d1.list)

{

ind.GetCheckListRentedAndAvailableVehicle();

}

foreach (var ind in d1.q\_list)

{

ind.GetCheckListRentedAndAvailableVehicle();

}

foreach (var ind in d1.d\_list)

{

ind.GetCheckListRentedAndAvailableVehicle();

}

foreach (var ind in d1.m\_list)

{

ind.GetCheckListRentedAndAvailableVehicle();

}

Console.WriteLine("\n");

foreach (var ind in d1.list)

{

ind.GetCheckListVehiclesInmaintanance();

}

foreach (var ind in d1.q\_list)

{

ind.GetCheckListVehiclesInmaintanance();

}

foreach (var ind in d1.d\_list)

{

ind.GetCheckListVehiclesInmaintanance();

}

foreach (var ind in d1.m\_list)

{

ind.GetCheckListVehiclesInmaintanance();

}

Console.WriteLine("\n");

foreach (var ind in d1.list)

{

ind.ShowAvailabilityforbookingforgivendate();

}

foreach (var ind in d1.q\_list)

{

ind.ShowAvailabilityforbookingforgivendate();

}

foreach (var ind in d1.d\_list)

{

ind.ShowAvailabilityforbookingforgivendate();

}

foreach (var ind in d1.m\_list)

{

ind.ShowAvailabilityforbookingforgivendate();

}

}

OutPut:

