**P-4**

**Interface and multithreading**

1.W.A.P. to implement following inheritance.

**Car(Abstract Class)**

Carno-

Name-

Abstract getCarInfo()

**Rate (Interface)**

Rate/km=7.5

Rate/day=500

getDays()

getKm()

**Customer**

calRate()

Display all info about Customer,Car,Km,Days and Total Amount.

2.W.A.P. having two threads one thread display Alphabets @ every 2 seconds and another thread display numbers from 1 to 20 @ every 1 second.

InterfaceDemo

using System;

using System.Collections.Generic;

using System.Text;

namespace InterfaceDemo

{

interface IEmployee

{

string Name

{ get;

set;

}

int Counter

{ get;

}

}

public class Employee : IEmployee

{ public static int numberOfEmployees;

private int counter;

private string name;

// Read-write instance property:

public string Name

{

get

{ return name;

}

set

{ name = value;

}

}

// Read-only instance property:

public int Counter

{

get

{ return counter;

}

}

// Constructor:

public Employee()

{ counter = ++counter + numberOfEmployees;

}

}

class Program

{

static void Main(string[] args)

{

Console.Write("Enter number of employees: ");

string s = Console.ReadLine();

Employee.numberOfEmployees = int.Parse(s);

Employee e1 = new Employee();

Console.Write("Enter the name of the new employee: ");

e1.Name = Console.ReadLine();

Console.WriteLine("The employee information:");

Console.WriteLine("Employee number: {0}", e1.Counter);

Console.WriteLine("Employee name: {0}", e1.Name);

}

}

}

Thread Demo

using System;

using System.Collections.Generic;

using System.Text;

using System.Threading;

namespace ThreadDemo

{

class Program

{public static void CallToChildThread()

{

try

{

Console.WriteLine("Child thread starts");

// do some work, like counting to 10

for (int counter = 0; counter <= 10; counter++)

{

Thread.Sleep(500);

Console.WriteLine(counter);

}

Console.WriteLine("Child Thread Completed");

}

catch (ThreadAbortException e)

{

Console.WriteLine("Thread Abort Exception");

}

finally

{

Console.WriteLine("Couldn't catch the Thread Exception");

}

}

static void Main(string[] args)

{

ThreadStart childref = new ThreadStart(CallToChildThread);

Console.WriteLine("In Main: Creating the Child thread");

Thread childThread = new Thread(childref);

childThread.Start();

//stop the main thread for some time

Thread.Sleep(2000);

//now abort the child

Console.WriteLine("In Main: Aborting the Child thread");

childThread.Abort();

Console.ReadKey();

}

}

}