1. Write a program to perform following interface.

**Car(Abstract Class)**

Carno-

Name-

Abstract getCarInfo()

**Customer**

calRate()

**Rate (Interface)**

Rate/km=7.5

Rate/day=500

getDays()

getKm()

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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace P401

{

public abstract class CarDetails

{

public String carname;

public String carnumber;

abstract public void Getcarinfo(string carname, string carnumber);

}

public interface RateDetails

{

void getdays();

void getkm();

}

class Program : CarDetails, RateDetails

{

private float rate\_km = 7.7f;

private float rate\_day = 500;

public int getday;

public int getkm;

public override void Getcarinfo(string carname, string carnumber)

{

String Carname = carname;

String Carnumber = carnumber;

Console.WriteLine("Car Name : {0} ", Carname);

Console.WriteLine("Car Number : {0} ", Carnumber);

}

public void get\_day()

{

Console.WriteLine("Enter Number of days");

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

//Console.WriteLine(rate\_day);

}

public void get\_km()

{

Console.WriteLine("Enter Killo Meter");

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

}

public void getdays()

{

throw new NotImplementedException();

}

void RateDetails.getkm()

{

throw new NotImplementedException();

}

public void calrate()

{

float ratekm = rate\_km \* getkm;

float rateday = rate\_day \* getday;

float total = ratekm + rateday;

Console.WriteLine("Total Km Rate : {0} ", ratekm);

Console.WriteLine("Total Day Rate : {0} ", rateday);

Console.WriteLine("Total Amount of Journy : {0} ", total);

}

static void Main(string[] args)

{

Program c = new Program();

c.Getcarinfo("Hundai", "GJ33A0300");

c.get\_day();

c.get\_km();

c.calrate();

Console.ReadKey();

}

}

}

O/P:-



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.

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 (char i ='A'; i <= 'Z'; i++)

{

Console.WriteLine(i);

Thread.Sleep(2000);

}

Console.WriteLine("Alphabeat Thread Complete");

}

catch (ThreadAbortException e)

{

Console.WriteLine("Thread Abort Exception");

}

finally

{

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

}

}

public static void Call()

{

try

{

Console.WriteLine("Child thread starts");

// do some work, like counting to 10

for (int i = 1; i <= 20; i++)

{

Console.WriteLine(i);

Thread.Sleep(1000);

}

Console.WriteLine("Number Thread Complete");

}

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);

Thread childThread = new Thread(childref);

ThreadStart ch= new ThreadStart(Call);

Thread child= new Thread(ch);

childThread.Start();

child.Start();

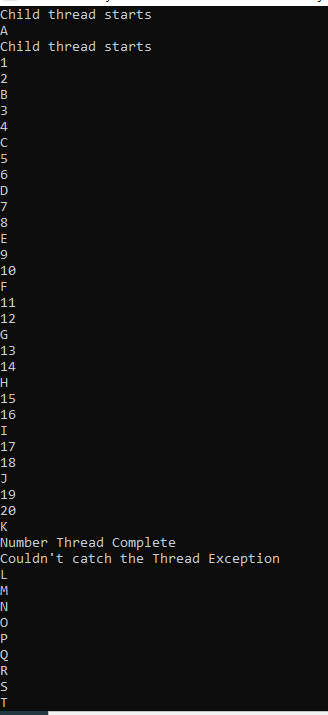
Console.ReadKey();

}

}

}

O/P:-



3 Write a c# program to perform stack operation.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Collections;

namespace GCG

{

class GCG

{

static void Main(string[] args)

{

Stack my\_stack = new Stack();

my\_stack.Push("Yash");

my\_stack.Push("Patel");

my\_stack.Push('A');

my\_stack.Push(null);

my\_stack.Push(1234);

my\_stack.Push(490.98);

my\_stack.Pop();

foreach (var elem in my\_stack)

{

Console.WriteLine(elem);

}

if (my\_stack.Contains(1234) == true)

{

Console.WriteLine("Element is found...!!");

}

else

{

Console.WriteLine("Element is not found...!!");

}

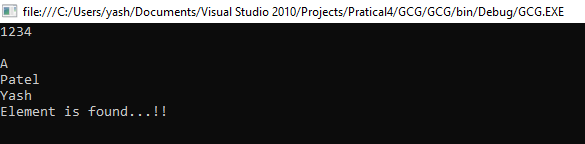
Console.ReadKey();

}

}

}

O/P:-



4 Write a c# program to perform queue operation.

using System;

using System.Collections;

class GFG

{

public static void Main()

{

Queue myQueue = new Queue();

myQueue.Enqueue("one");

Console.Write("Total number of elements in the Queue are : ");

Console.WriteLine(myQueue.Count);

myQueue.Enqueue("two");

Console.Write("Total number of elements in the Queue are : ");

Console.WriteLine(myQueue.Count);

myQueue.Enqueue("three");

Console.Write("Total number of elements in the Queue are : ");

Console.WriteLine(myQueue.Count);

myQueue.Enqueue("four");

Console.Write("Total number of elements in the Queue are : ");

Console.WriteLine(myQueue.Count);

myQueue.Enqueue("five");

Console.Write("Total number of elements in the Queue are : ");

Console.WriteLine(myQueue.Count);

myQueue.Enqueue("six");

Console.Write("Total number of elements in

the Queue are : ");

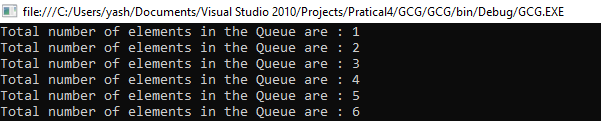
Console.WriteLine(myQueue.Count);

Console.ReadKey();

}

}

O/P:-



5 Write a c# program to perform array List.

using System;

using System.Collections;

class GFG

{

public static void Main()

{

ArrayList myArryList = new ArrayList();

myArryList.Add(1);

myArryList.Add("Two");

myArryList.Add(3);

myArryList.Add(4.5f);

int firstElement = (int)myArryList[0];

string secondElement = (string)myArryList[1];

int thirdElement = (int)myArryList[2];

float fourthElement = (float)myArryList[3];

myArryList.Remove(100);

foreach (var item in myArryList)

Console.WriteLine(item);

Console.ReadKey();

}

}

O/P:-

