**Date:12/04/2024  
  
Task : Creation of doctor**  
  
namespace day12\_tasks

{

internal class Program

{

static void Main(string[] args)

{

doctors createdoctor(int id)

{

doctors doc = new doctors(id);

Console.WriteLine($"Please enter the Doctor{id} name ");

doc.Name = Console.ReadLine();

int Age;

Console.WriteLine($"Please enter the Doctor\_{id} Age ");

while (!int.TryParse(Console.ReadLine(), out Age))

{

Console.WriteLine("Please enter valid Age:");

}

doc.Age = Age;

Console.WriteLine($"Please enter the Doctor{id} Qualification ");

doc.Qualification = Console.ReadLine();

Console.WriteLine($"Please enter the Doctor{id} Experience ");

int experience;

while (!int.TryParse(Console.ReadLine(), out experience))

{

Console.WriteLine("Please enter valid Experience:");

}

doc.Experience = experience;

Console.WriteLine($"Please enter the Doctor{id} Speciality ");

doc.Speciality = Console.ReadLine();

return doc;

}

Console.WriteLine("Enter How many doctors do you want to create:");

int no\_doctor=int.Parse(Console.ReadLine());

doctors[] doctors = new doctors[no\_doctor];

for (int i = 0; i < no\_doctor; i++)

{

doctors[i]=createdoctor(i+1);

}

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.WriteLine("Here is the doctors List :");

for (int i = 0; i < no\_doctor; i++)

{

doctors[i].PrintdoctorsDetails();

}

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.WriteLine("Enter Speciality :");

string spec = Console.ReadLine();

for (int i = 0; i < no\_doctor; i++)

{

doctors[i].PrintdoctorsDetails(spec);

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace day12\_tasks

{

internal class doctors

{

public int Id { get; set; }

public String Name { get; set; }

public int Age { get; set; }

public int Experience { get; set; }

public String Qualification { get; set; }

public String Speciality { get; set; }

public doctors(int id ) {

Id = id;

}

public void PrintdoctorsDetails(string? Spec = null)

{

if (Spec == null || Spec == Speciality.ToString())

{

Console.WriteLine($"doctor Id\t:\t{Id}");

Console.WriteLine($"doctor name\t:\t{Name}");

Console.WriteLine($"doctor Age \t:\t{Age} years");

Console.WriteLine($"doctor Qualification\t:\t.{Qualification}");

Console.WriteLine($"doctor Speciality\t:\t{Speciality}");

Console.WriteLine($"doctor Experience\t:\t{Experience} years");

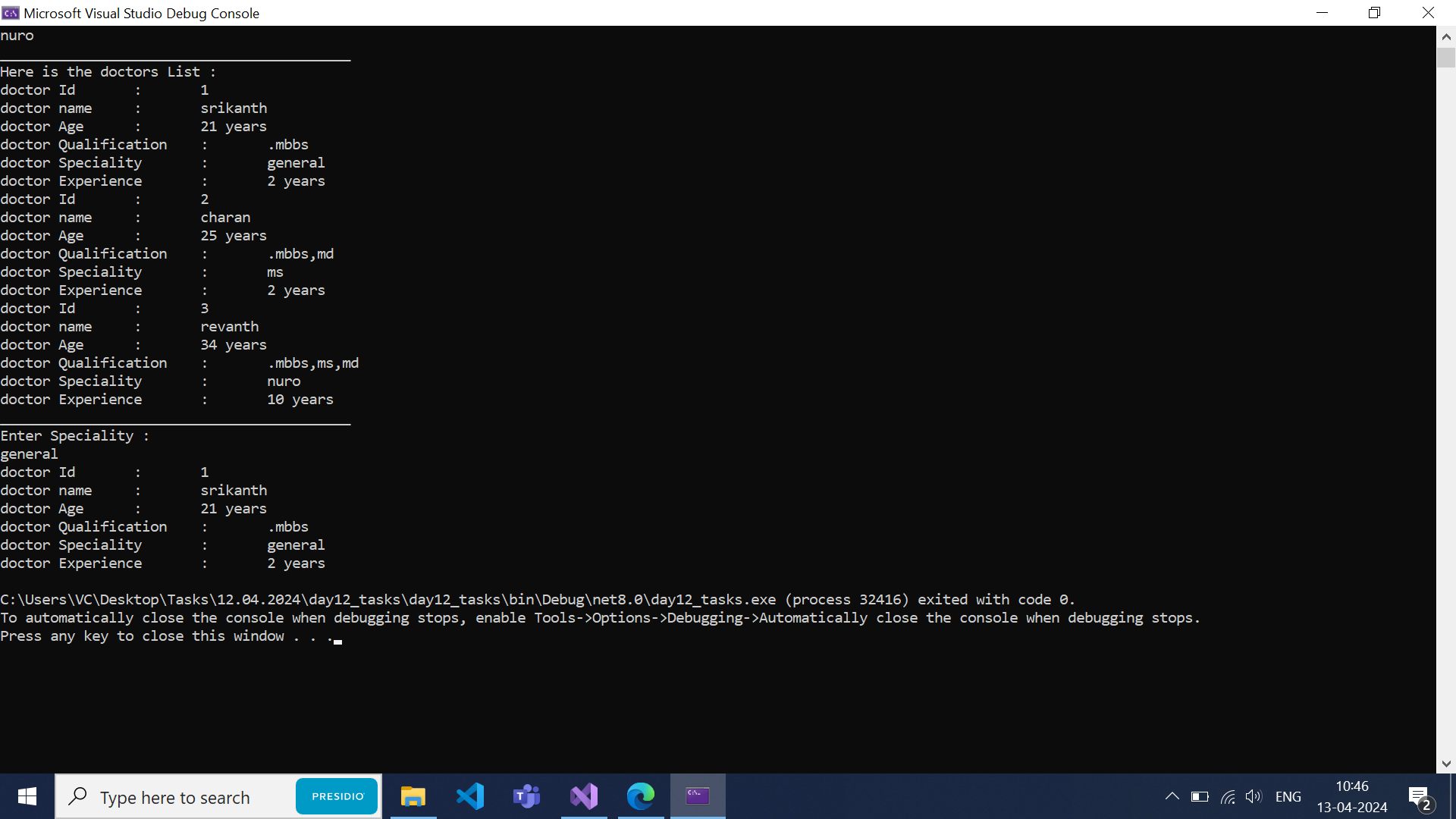
}

}

}

}

**Output:**

  
  
  
**task 2:Card validation**

using static System.Runtime.InteropServices.JavaScript.JSType;

namespace card\_validation

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter a card number");

string card=Console.ReadLine();

int[] nums = new int[16];

for (int i = 0; i < 16; i++)

{

while (!int.TryParse(card[i].ToString(), out nums[i]) || card.Length != 16)

{

Console.WriteLine("Please enter a valid 16-digit card number:");

card = Console.ReadLine();

}

}

int sum = 0;

Array.Reverse(nums);

for (int i = 0; i < 16; i++)

{

nums[i] = i % 2 != 0 ?nums[i] \* 2:nums[i];

nums[i] = nums[i] > 9 ? nums[i] - 9 : nums[i];

sum += nums[i];

}

if (sum%10==0) Console.WriteLine("Yess this a valid card number:");

else Console.WriteLine("Sorry this a Invalid card number:");

}

}

}

Output