#### Taks\_1 ) Create a Doctor class with details Id, Name, Age, Exp, Qualification, Speciality, Create an array of doctors ,Print the array, Given a speciality print the doctor details in it.

Ans)

Programe.cs :

using System.Transactions;

namespace Task\_1

{

internal class Program

{

/// <summary>

/// Handling Input , Exception for Integer feild

/// </summary>

/// <returns>Integer</returns>

int HandlingIntegerInput()

{

int num = 0;

while(int.TryParse(Console.ReadLine(),out num) == false)

{

Console.WriteLine("Please Enter a valid number \n");

}

return num;

}

/// <summary>

/// Handling Input , Exception for String feild

/// </summary>

/// <returns>String</returns>

string HandlingStringInput()

{

string ? str =string.Empty;

do

{

str = Console.ReadLine();

if (string.IsNullOrWhiteSpace(str))

Console.WriteLine("Please Enter a valid string \n");

} while (string.IsNullOrWhiteSpace(str));

return str;

}

/// <summary>

/// Takes Counter (Integer) as Argument and Creats Doctor

/// </summary>

/// <param name="counter"></param>

/// <returns>Doctor class</returns>

Doctor CreateDoctor(int counter) {

int id = int.Parse((DateTime.Now.ToString("yyyyMMdd"))),age ,experience;

string name, qualification, speciality;

Console.WriteLine($"Please Enter the Age of {counter + 1} doctor : ");

age = HandlingIntegerInput();

Console.WriteLine($"Please Enter the Experience of {counter + 1} doctor : ");

experience = HandlingIntegerInput();

Console.WriteLine($"Please Enter the Name of {counter + 1} doctor : ");

name = HandlingStringInput();

Console.WriteLine($"Please Enter the Qualification of {counter + 1} doctor : ");

qualification = HandlingStringInput();

Console.WriteLine($"Please Enter the Speciality of {counter + 1} doctor : ");

speciality = HandlingStringInput();

Doctor doctor = new Doctor(id,name,age,experience,qualification,speciality);

return doctor;

}

/// <summary>

/// Takes Specailty (String) , Doctor (Array)

/// </summary>

/// <param name="specailty"></param>

/// <param name="doctors"></param>

static void GetDoctorBySpecailty(string specailty , Doctor[] doctors)

{

bool flag = false;

for(int i = 0; i < doctors.Length; i++)

{

if (doctors[i].Speciality.ToLower() == specailty.ToLower())

{

flag = true;

doctors[i].PrintingDoctorDetails();

break;

}

}

if(!flag) { Console.WriteLine("There Are No Such Doctors With Given Speciality"); }

return;

}

static void Main(string[] args)

{

Program program = new Program();

Console.WriteLine("Please Enter the number of doctors : ");

//Console.WriteLine(DateTime.Now.ToString("yyyyMMdd"));

int size = program.HandlingIntegerInput();

Doctor[] doctors = new Doctor[size];

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

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

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n");

Console.WriteLine($"Please Enter Doctor {i+1} Details \n");

doctors[i] = program.CreateDoctor(i);

}

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Fetching All Doctors \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

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

{

Console.WriteLine($"The Doctor {i + 1} Details \n");

doctors[i].PrintingDoctorDetails();

}

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Search Doctor by Speciality\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

string speciality;

do

{

Console.WriteLine("Please Enter the Speciality of Doctor or Enter q for quit: ");

speciality = program.HandlingStringInput();

if(speciality != "q") GetDoctorBySpecailty(speciality, doctors);

} while (speciality != "q");

}

}

}

**Doctor.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task\_1

{

internal class Doctor

{

/// <summary>

/// creating a id (Integer) and Getting it up

/// </summary>

public int Id { get; private set; }

/// <summary>

/// Creating a name (String) and Getting it up

/// </summary>

public string Name { get; private set; }

/// <summary>

/// Creating a Age(Integer) and Getting it up

/// </summary>

public int Age { get; private set; }

/// <summary>

/// Creating a Experience(Integer) and Getting it up

/// </summary>

public int Experience { get; private set; }

/// <summary>

/// Creating a Qualification(String) and Getting it up

/// </summary>

public string Qualification { get; private set; }

/// <summary>

/// Creating a Speciality(String) and Getting it up

/// </summary>

public string Speciality { get; private set; }

/// <summary>

/// Default Constructor for the Class : Doctor

/// </summary>

public Doctor()

{

Id = 0;

Name = string.Empty;

Age = 0;

Experience = 0;

Qualification = string.Empty;

Speciality = string.Empty;

}

/// <summary>

/// Parameterized Constructor for Class : Doctor

/// </summary>

/// <param name="id"></param>

/// <param name="name"></param>

/// <param name="age"></param>

/// <param name="experience"></param>

/// <param name="qualification"></param>

/// <param name="speciality"></param>

public Doctor(int id, string name, int age , int experience, string qualification, string speciality)

{

Id = id;

Name = name;

Age = age;

Experience = experience;

Qualification = qualification;

Speciality = speciality;

}

public void PrintingDoctorDetails()

{

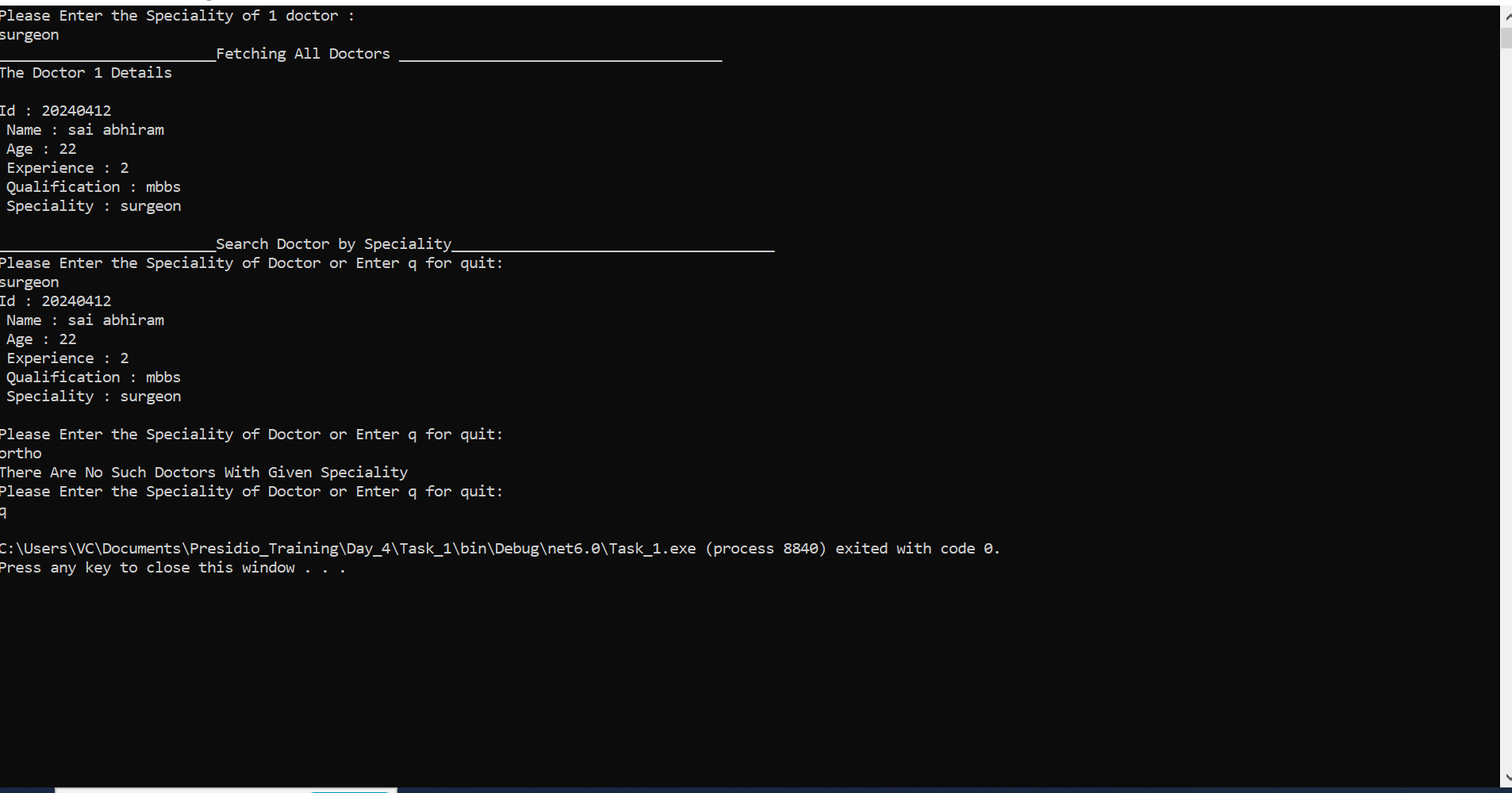
Console.WriteLine($"Id : {Id} \n Name : {Name} \n Age : {Age} \n Experience : {Experience} \n Qualification : {Qualification} \n Speciality : {Speciality} \n");

}

}

}

**Output :**



Task\_2) Acess Card Validator

Ans)

**Programe.cs:**

namespace Taks\_2

{

internal class Program

{

/// <summary>

/// Handle the String input and Perform Validation

/// </summary>

/// <returns>String</returns>

string HandlingStringInput()

{

string? str;

do

{

str = Console.ReadLine();

if(str?.Length != 16 || string.IsNullOrEmpty(str))

{

Console.WriteLine("Please Enter a Valid Card Number");

}

} while (string.IsNullOrEmpty(str) || str?.Length != 16);

return str;

}

/// <summary>

/// Contains the Controller logic for CardNumber

/// </summary>

/// <param name="str"></param>

/// <returns></returns>

static bool check(string str)

{

string ReverseStr = new string(str.Reverse().ToArray());

int sum = 0;

for(int i = 0; i < ReverseStr.Length; i++)

{

if((i+1) % 2 == 0)

{

int temp = ((int)ReverseStr[i] - 48);

temp = temp \* 2;

sum += (temp % 10) + (temp / 10);

}

else

{

sum+= ((int)ReverseStr[i]-48);

}

}

return sum%10 == 0;

}

static void Main(string[] args)

{

Program program = new Program();

Console.WriteLine("Please Enter Card Details (size of card should be 16) : ");

string CardNumber = program.HandlingStringInput();

if(check(CardNumber) )

{

Console.WriteLine("The Card is Valid");

}

else

{

Console.WriteLine("The Card is Invalid");

}

}

}

}

**Output:**

