Barani Institute of Information Technology

NAME	:	TAYYABA KOUSAR
CLASS	:	BSCS-4A
REG NO	:	2024-Arid-0200
SUBMITTED TO	:	SIR AFTAB
ASSINGMENT N0	:	1

QUESTION 1:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace assgnom1
{
    internal class Program
    {
        static List<Product> products = new List<Product>();
        static void Main(string[] args)

        {
            int choice;



            do
            {
                Console.WriteLine("Press 1 Add product");
                Console.WriteLine("press 2 update product");
                Console.WriteLine("press 3 Remove product");
                Console.WriteLine("press 4 search product by ID");
                Console.WriteLine("press 5 search product by price
range");
                Console.WriteLine("press 6 Display All products ");
                Console.WriteLine("Exit");
                choice=int.Parse(Console.ReadLine());
                switch(choice){
                    case 1:
                        AddProduct();
                        break;
                    case 2:
                        UpdateProduct();
                        break;
                    case 3:
                        RemoveProduct();
                        break;
                    case 4:
                        SearchByID();
                        break;
                    case 5:
                        SearchByPriceRange();
                        break;
                    case 6:
                        DispalyAll();
                        break;
                    case 7:
                        Console.WriteLine("Exiting Program!!");
                        break;
                    default:
                        Console.WriteLine("Invalid choice");
                        break;

                }


            } while (choice!=7);
        }

        static void AddProduct()
        {
            Product p=new Product();
            Console.WriteLine("Enter Product Id");
            p.productid=int.Parse(Console.ReadLine());
            Console.WriteLine("Enter Product Name");
            p.pname = Console.ReadLine();
            Console.WriteLine("Enter Product category");
```

```
   p.category = Console.ReadLine();
   Console.WriteLine("Enter Product price");
   p.price = int.Parse(Console.ReadLine());
   Console.WriteLine("Enter Product stock quantity");
   p.quantity = int.Parse(Console.ReadLine());
   Console.WriteLine("Enter Product company name");
   p.companyname =Console.ReadLine();
   products.Add(p);

}
static void UpdateProduct() {
   Console.WriteLine("Enter Product Id");
   int id = int.Parse(Console.ReadLine());

 Product found=products.Find(p=>p.productid==id);
   if (found != null) {
      Console.WriteLine("Enter Product Name");
      found.pname = Console.ReadLine();
      Console.WriteLine("Enter Product category");
      found.category = Console.ReadLine();
      Console.WriteLine("Enter Product price");
      found.price = int.Parse(Console.ReadLine());
      Console.WriteLine("Enter Product stock quantity");
      found.quantity = int.Parse(Console.ReadLine());
      Console.WriteLine("Enter Product company name");
      found.companyname = Console.ReadLine();
   }
   else
   {
      Console.WriteLine("Product not found");
   }

}
static void RemoveProduct() {
   Console.WriteLine("Enter Product Id");
   int id = int.Parse(Console.ReadLine());
```

```
            Product found = products.Find(p => p.productid == id);
            if (found != null)
            {
                products.Remove(found);
                Console.WriteLine("product is sucessfully removed");
            }
            else
            {
                Console.WriteLine("Product not found");
            }
        }
        static void SearchByID() {
            Console.WriteLine("Enter Product Id");
            int id = int.Parse(Console.ReadLine());

            Product found = products.Find(p => p.productid == id);
            if (found != null)
            {
                Console.WriteLine(" Product Id" + found.productid);
                Console.WriteLine(" Product Name"+ found.pname);
                Console.WriteLine(" Product category"+ found.category);
                Console.WriteLine(" Product price" + found.price);
                Console.WriteLine("Product stock quantity" +
found.quantity);
                Console.WriteLine(" Product company name" +
found.companyname);

            }
            else
            {
                Console.WriteLine("Product not found");
            }
        }
        static void SearchByPriceRange() {
            Console.WriteLine("Enter Product max price");
            int max = int.Parse(Console.ReadLine());
            Console.WriteLine("Enter Product min price");
            int min = int.Parse(Console.ReadLine());
            bool foundany=false;
            foreach (Product p in products)
            {
                if(p.price >= min && p.price <= max)
                {
                    Console.WriteLine(" Product Id" + p.productid);
                    Console.WriteLine(" Product Name" + p.pname);
                    Console.WriteLine(" Product price" + p.price);

                }


            }
            if (!foundany)
            {
                Console.WriteLine("no products in rage");
            }

        }
        static void DispalyAll() {
            if (products.Count == 0)
            {
                Console.WriteLine("no product avalible");
            }
            else
            {
                foreach (Product p in products)
                {
                    Console.WriteLine(" Product Id" + p.productid);
                    Console.WriteLine(" Product Name" + p.pname);
                    Console.WriteLine(" Product category" + p.category);
                    Console.WriteLine(" Product price" + p.price);
                    Console.WriteLine("Product stock quantity" +
p.quantity);
                    Console.WriteLine(" Product company name" +
p.companyname);
```

```
                }
            }
        }

    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace assgnom1
{
    internal class Product
    {
        public int productid {  get; set; }
        public string pname { get; set; }
        public string category { get; set; }
        public int price { get; set; }

        public int quantity { get; set; }
        public string companyname{ get; set; }
    }
}
```

QUESTION 2:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace StudentCourseMng
{

    internal class Student
    {
        public int StudentId { get; set; }
        public string Name { get; set; }
        public DateTime DateOfBirth { get; set; }
        public string Gender { get; set; }
        public string City { get; set; }
        public string Department { get; set; }
        public string Course { get; set; }
        public bool IsMarried { get; set; }
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Xml.Linq;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace StudentCourseMng
{
    public partial class Form1 : Form
    {   List <Student> students = new List <Student> ();
        Dictionary<string, List<string>> course = new Dictionary<string,
List<string>>()
            {
                { "CS", new List<string> { "advance programing", "Coal" } },
                { "AI", new List<string> { "MLP", "PIA" } },
                 { "SE", new List<string> { "SRE", "SEO" } },
                 { "Math", new List<string> { "LA, CGA" } }
            };
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            string[] cities = new string[]
            {
                "Lahore", "Faislabad",
"Multan","Karachi","Hyderabad","sukkar"
            };
            combCity.DataSource=cities;

            List<String> Department = new List<String>() {
                "CS" , "AI" ,"SE","Math"
            };
            comboDep.DataSource=Department;
            string selectedDepartment = comboDep.SelectedItem.ToString();
            combocourse.DataSource = course[selectedDepartment];


        }


        private void insert_Click(object sender, EventArgs e)
        {
            int id = int.Parse(txtstu.Text);

            if (students.Any(s => s.StudentId == id))
            {
                MessageBox.Show("Student ID already exists!");
                return;
            }
            string gender = radioButton1.Checked ? "Male" : "Female";

            Student student = new Student()
            {
                StudentId = id,
                Name = textName.Text,
                DateOfBirth = dateTimePicker1.Value,
                Gender = gender,
                IsMarried = checkBox1.Checked,
                City = combCity.Text,
                Department = comboDep.Text,
                Course = combocourse.Text
            };

            students.Add(student);
            MessageBox.Show("Student added successfully!");
        }

        private void Update_Click(object sender, EventArgs e)
        {
           int id = int.Parse(txtstu.Text);
            Student student = students.Find(s => s.StudentId == id);

            if (student != null)
            {
                student.Name = textName.Text;
                student.DateOfBirth = dateTimePicker1.Value;
                student.Gender = radioButton1.Checked ? "Male" : "Female";
                student.IsMarried = checkBox1.Checked;
                student.City = combCity.Text;
                student.Department = comboDep.Text;
                student.Course = combocourse.Text;

                MessageBox.Show("Student updated successfully!");
            }
            else
            {
                MessageBox.Show("Student not found!");
            }
        }

        private void Delete_Click(object sender, EventArgs e)
        {
            int id = int.Parse(txtstu.Text);
            Student student = students.Find(s => s.StudentId == id);

            if (student != null)
            {
                students.Remove(student);
                MessageBox.Show("Student deleted successfully!");
            }
            else
            {
                MessageBox.Show("Student not found!");
            }
        }

        private void Exit_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void comboDep_SelectedIndexChanged(object sender, EventArgs
e)
        {
            if (comboDep.SelectedItem == null) return;
            string selectedDepartment = comboDep.SelectedItem.ToString();

            if (course.ContainsKey(selectedDepartment))
            {
                combocourse.DataSource = null;
                combocourse.DataSource = course[selectedDepartment];
            }
        }
    }
}
```