1. Write a program to show the concept of 3 dimensional array**.**

Using System;

namespace Array{

public class Program{

staic void Main(string []args){

int[,,] arr =

{

{ {1, 3, 5}, {2, 4, 6} },

{ {1, 3, 5}, {2, 4, 6} }

};

Console.WriteLine("Elements of the 3D array:");

for (int i = 0; i < arr.GetLength(0); i++)

{

for (int j = 0; j < arr.GetLength(1); j++)

{

for (int k = 0; k < arr.GetLength(2); k++)

{

Console.WriteLine("arr[{0}, {1}, {2}] = {3}", i, j, k, arr[i, j, k]);

}

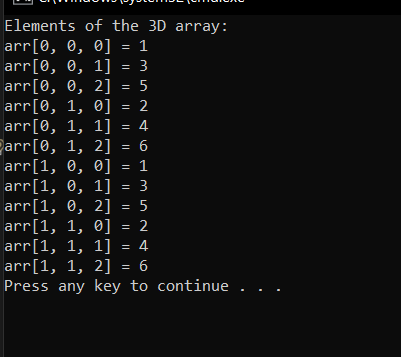
}

}

}

}

}  
output:



1. WAP to show the example of using method hiding and method overriding.

For method overriding:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace hideOverride

{

class parent

{

public virtual void Show()

{

Console.WriteLine("Im parent");

}

}

class child : parent

{

public override void Show()

{

Console.WriteLine("IM child and came from override");

}

}

internal class Program

{

static void Main(string[] args)

{

parent p= new child();

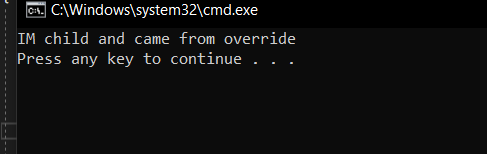
p.Show();

}

}

}

Output:



For method hiding:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace hideOverride

{

class parent

{

public void Show()

{

Console.WriteLine("Im parent");

}

}

class child : parent

{

public new void Show()

{

Console.WriteLine("IM child and came from override");

}

}

internal class Program

{

static void Main(string[] args)

{

parent p= new child();

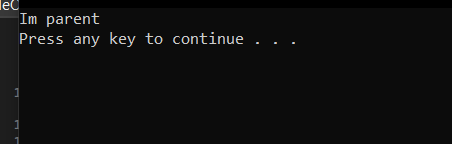
p.Show();

}

}

}

Output:



3. WAP to show the exmaple of using "enum" and "structs".**X**

namespace EnumStruct

{

enum level { low,middle,high};

struct student

{

public int id;

public string name;

};

internal class Program

{

static void Main(string[] args)

{

level l = level.low;

int s= (int)level.high;

Console.WriteLine(s);

Console.WriteLine(l);

student s1= new student();

s1.id = 1;

s1.name = "Bibek Karki";

Console.WriteLine(s1.id);

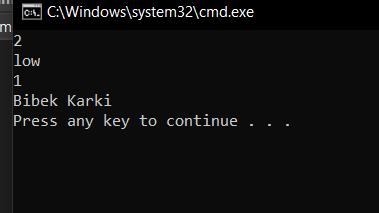
Console.WriteLine(s1.name);

}

}

}

Output:



1. WAP to show the example of File handling (create Folder, Files, select files and folders).

using System;

using System.Collections.Generic;

using System.Globalization;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FileHandling

{

internal class Program

{

static void Main(string[] args)

{

string rootFolder = @"C:\Users\nawar\Downloads";

Directory.CreateDirectory(rootFolder);

string filePath = Path.Combine(rootFolder, "example.txt");

File.WriteAllText(filePath, "Im student of KCT.");

Console.WriteLine("Folder created: " + rootFolder);

Console.WriteLine("File created: " + filePath);

Console.WriteLine("Enter a folder path:");

string selectedFolder = Console.ReadLine();

if (Directory.Exists(selectedFolder))

{

Console.WriteLine("Selected folder: " + selectedFolder);

}

else

{

Console.WriteLine("Invalid folder path!");

}

Console.WriteLine("Enter a file path:");

string selectedFile = Console.ReadLine();

if (File.Exists(selectedFile))

{

Console.WriteLine("Selected file: " + selectedFile);

string data=File.ReadAllText(selectedFile);

Console.WriteLine(data);

}

else

{

Console.WriteLine("Invalid file path!");

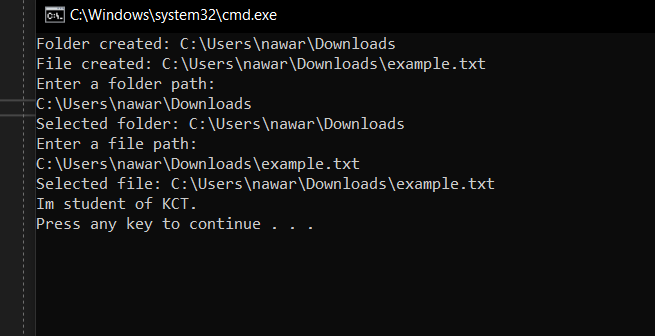
}

}

}

}

Output:



1. WAP to show the even, square of list of numbers using Lambda expression.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lamda

{

internal class Program

{

static void Main(string[] args)

{

List<int> numbers = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

var evenNumbers = numbers.FindAll(num => num % 2 == 0);

var squares = numbers.Select(num => num \* num);

Console.WriteLine("Even Numbers:");

foreach (var num in evenNumbers)

{ Console.WriteLine(num);

}

Console.WriteLine("Squares of Numbers:");

foreach (var square in squares)

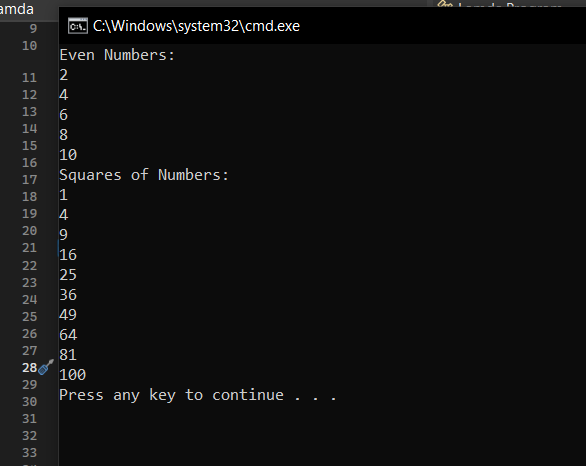
{

Console.WriteLine(square);

}

} }}

Output:



1. WAP to add any five students detail and find the records whose address is lalitpur and college is kct and order by firstname descending in LINQ.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LINQ

{

internal class Program

{

class Student

{

public string FirstName { get; set; }

public string LastName { get; set; }

public string Address { get; set; }

public string College { get; set; }

}

static void Main(string[] args)

{

List<Student> students = new List<Student>();

students.Add(new Student { FirstName = "Saksham", LastName = "Khanal", Address = "Lalitpur", College = "KCT" });

students.Add(new Student { FirstName = "Bibke", LastName = "Karki", Address = "Kathmandu", College = "KCT" });

students.Add(new Student { FirstName = "Roshan", LastName = "sth", Address = "Lalitpur", College = "BKT campus" });

students.Add(new Student { FirstName = "Niusm", LastName = "Giri", Address = "Lalitpur", College = "KCT" });

students.Add(new Student { FirstName = "Sita", LastName = "Karki", Address = "Bhaktapur", College = "GGIC" });

var filteredStudents = from student in students

where student.Address == "Lalitpur" && student.College == "KCT"

orderby student.FirstName descending

select student;

Console.WriteLine("Students with Address 'Lalitpur' and College 'KCT' (Ordered by First Name Descending):");

foreach (var student in filteredStudents)

{

Console.WriteLine($"Name: {student.FirstName} {student.LastName}, Address: {student.Address}, College: {student.College}");

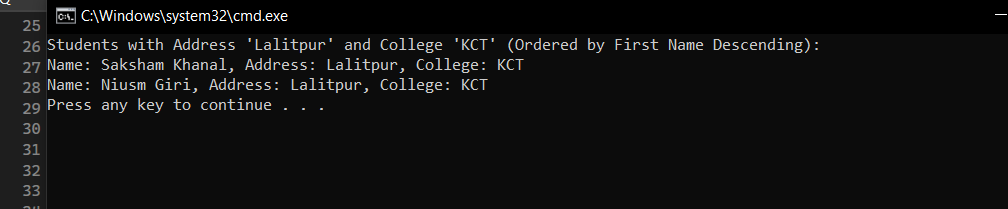
}

}

}

}

Output:



1. WAP to show the example of using abstract class and interface.

For abstract:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AbstractInter

{

abstract class Animal

{

public abstract void Sound();

public void Eat()

{

Console.WriteLine("Eat:True");

}

}

class Dog : Animal

{

public override void Sound()

{

Console.WriteLine("Bark");

}

}

internal class Program

{ static void Main(string[] args)

{

Dog d1 = new Dog();

d1.Eat();

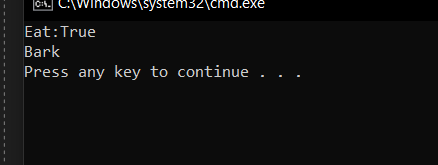
d1.Sound();

}

}

}

Output:



For interface:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AbstractInter

{

interface Vehicels

{ void Tires();

}

class Bike : Vehicels

{

public void Tires()

{

Console.WriteLine("Bike has got two tires");

}

}

internal class Program

{ static void Main(string[] args)

{

Bike b1 = new Bike();

b1.Tires();

}

}

}

Output:

