# Lab 1: Write a C# program to add two digit using Constructor

Code:

// Write a C# program to add two digit using Constructor

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab1\_milan\_26806

{

class add

{

public int sum;

public add(int x, int y)

{

sum = x + y;

}

public void value()

{

Console.WriteLine(sum);

}

}

internal class Program

{

static void Main(string[] args)

{

Console.Write("Enter the first number\t");

int a = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter the second number\t");

int b = Convert.ToInt32(Console.ReadLine());

add sum = new add(a, b);

Console.Write("The sum is:\n");

sum.value();

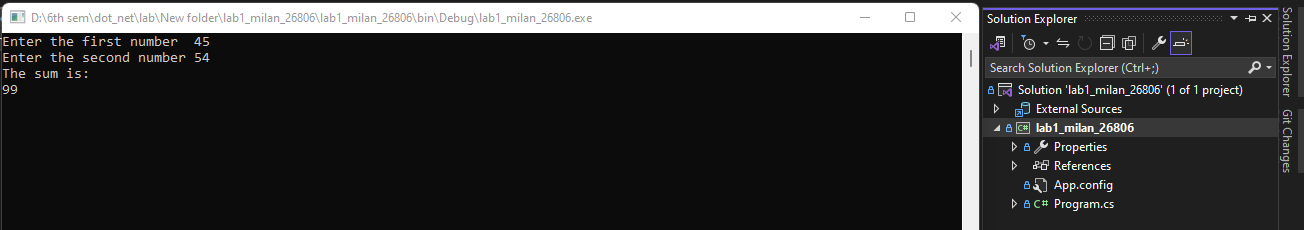
Console.ReadLine();

}

}

}

Output:



# LAB 2: Write a C# program to initialize and display jagged array elements with sum of each row.

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab2\_milan\_26806

{

internal class Program

{

static void Main(string[] args)

{

int count = 0;

int[][] arr2 = new int[3][];

arr2[0] = new int[] { 12, 35, 12, 51, 53 };

arr2[1] = new int[] { 32, 56, 12, 78, 54, 19, 87, 43, 26 };

arr2[2] = new int[] { 75, 59, 35, 46, 24, 16, 61 };

Console.WriteLine("Jagged array");

for (int i = 0; i < arr2.Length; i++)

{

for (int j = 0; j < arr2[i].Length; j++)

{

count = count + arr2[i][j];

Console.Write(arr2[i][j] + " ");

}

Console.WriteLine();

}

Console.WriteLine("The sum is:" + count);

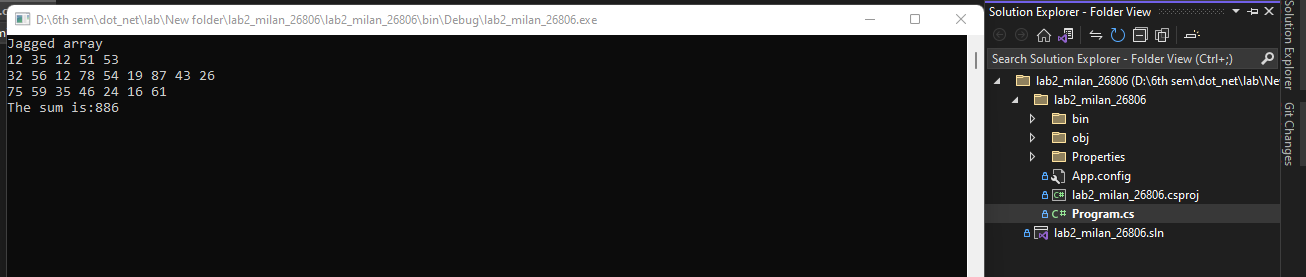
Console.ReadLine();

}

}

}

Output:



# LAB 3: Write a C# program to initialize and display 2D array elements with sum of each row.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab3\_milan\_26806

{

internal class Program

{

static void Main(string[] args)

{

int count = 0;

int[,] arr2 = new int[3, 3] { { 12, 35, 12 }, { 51, 53, 32 }, { 56, 12, 78 } };

Console.WriteLine("2D array");

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

count = count + arr2[i, j];

Console.Write(arr2[i, j] + " ");

}

Console.WriteLine();

}

Console.WriteLine("The sum is:" + count);

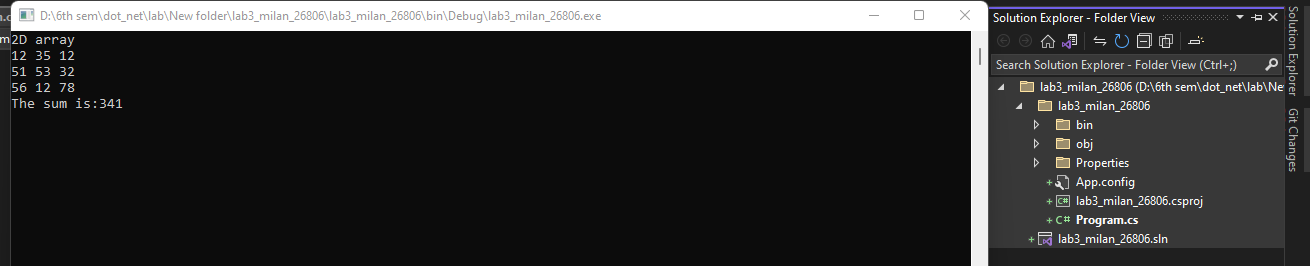
Console.ReadLine();

}

}

}

Output:



# Lab 4: Write a C# Program to calculate Area of Rectangle using Single Inheritance.

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab\_4\_milan\_26806

{

class shape

{

protected int width;

protected int height;

public void set(int x, int y)

{

width = x;

height = y;

}

}

class Rectangle : shape

{

public int area()

{

return width \* height;

}

}

internal class Program

{

static void Main(string[] args)

{

Console.Write("Enter the width of the rectangle\t");

int x = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter the height of the rectangle\t");

int y = Convert.ToInt32(Console.ReadLine());

Rectangle r = new Rectangle();

r.set(x, y);

Console.Write("The area is: {0}", r.area(), "\n");

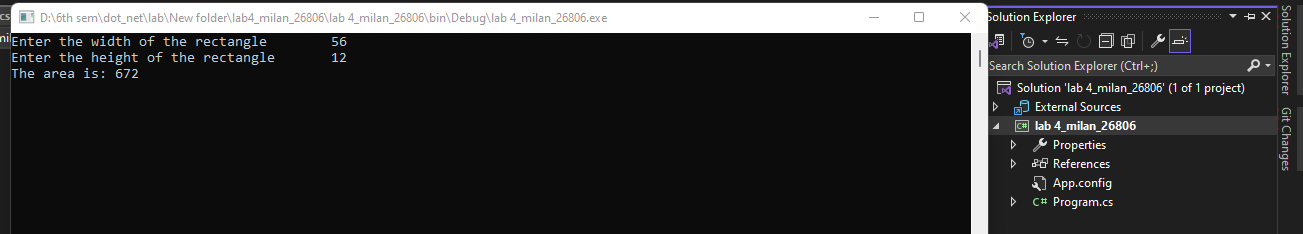
Console.Read();

}

}

}

Output:



# Lab 5: Write a C# program to calculate Area and Paint Cost of Rectangle using multiple Inheritance.

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab5\_milan\_26806

{

class shape

{

protected int width;

protected int height;

public void set(int x, int y)

{

width = x;

height = y;

}

}

public interface area

{

int getarea();

}

class cost : shape, area

{

public int a;

public int getarea()

{

a = width \* height;

return a;

}

public int costof(int ar, int price)

{

return a \* price;

}

}

internal class Program

{

static void Main(string[] args)

{

Console.Write("Enter the width of the rectangle\t");

int x = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter the height of the rectangle\t");

int y = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter the price of painting\t");

int price = Convert.ToInt32(Console.ReadLine());

cost r = new cost();

r.set(x, y);

int area = r.getarea();

Console.Write("The area is: {0}", area, "\n");

Console.Write("\nThe cost is: {0}", r.costof(area, price));

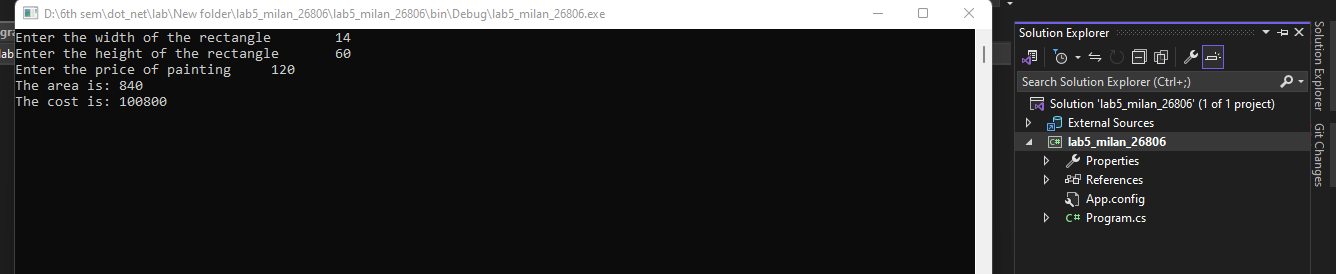
Console.Read();

}

}

}

Output:



Lab 6 .Write a C# Sharp program to find the position of a specified word in a given string.

Code:

//Write a C# Sharp program to find the position of a specified word in a given string.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab6\_milan\_26806

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter a sentence.");

string str = Console.ReadLine();

Console.WriteLine("Enter a word to find its position in the sentence.");

string word = Console.ReadLine();

string[] tokens = str.Split(' ');

int a = Array.IndexOf(tokens, word) + 1;

Console.WriteLine("The position of {0} is {1}", word, a);

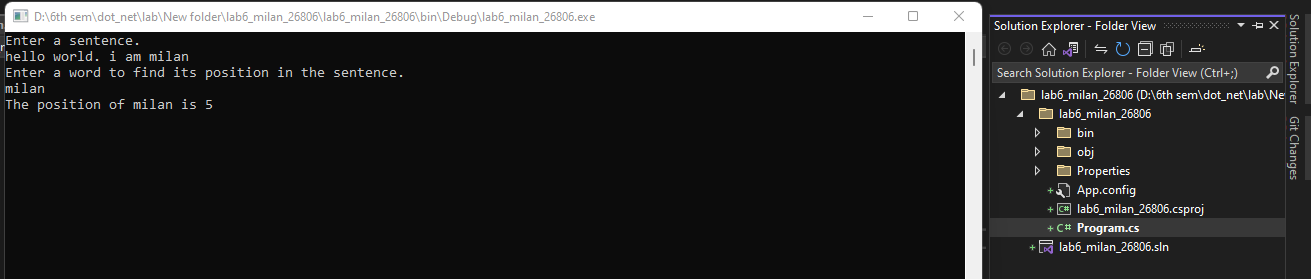
Console.ReadLine();

}

}

}

Output:



Lab 7. Write a program in C# Sharp to count the total number of words and Character in a string.

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab7\_milan\_26806

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter a sentence.");

string str = Console.ReadLine();

string[] tokens = str.Split(' ');

Console.Write("The number of words :{0}\nThe number of Characters:{1}", tokens.Length, str.Length);

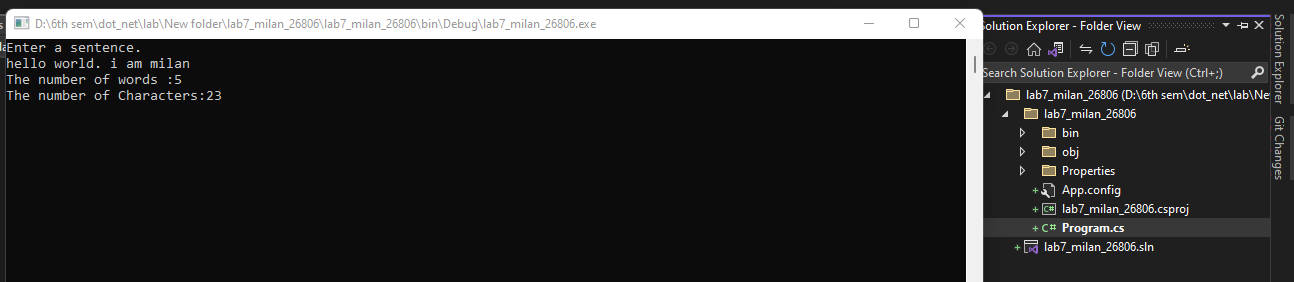
Console.ReadLine();

}

}

}

Output:



Lab 8. Write a program in C# Sharp to count the number of alphabets, digits and special characters in a string

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab8\_milan\_26806

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter a sentence.");

string str = Console.ReadLine();

int speccount = 0;

int alpcount = 0;

int digcount = 0;

foreach (char c in str)

{

if (char.IsLetter(c))

{

alpcount++;

}

else if (char.IsDigit(c))

{

digcount++;

}

else

{

speccount++;

}

}

Console.Write("The number of alphabets: {0}\nThe number of digits: {1}\nThe number of special characters: {2}", alpcount, digcount, speccount);

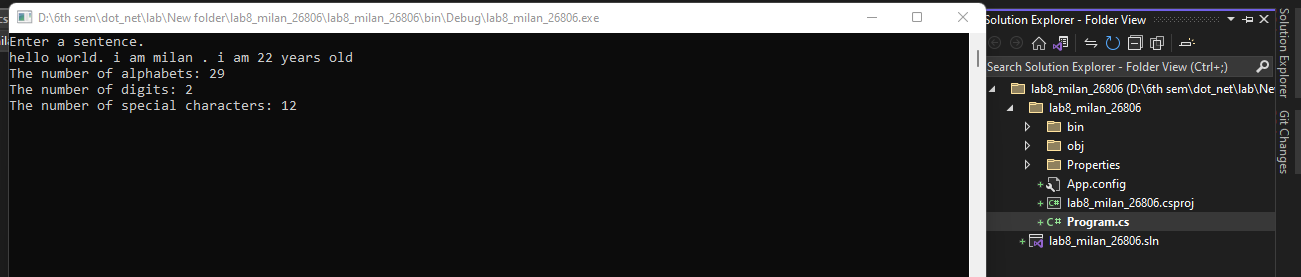
Console.ReadLine();

}

}

}

Output:



Lab 9. Write a C# Sharp program to count the number of vowels or consonants in a string.

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab9\_milan\_26806

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter a String.");

string str = Console.ReadLine();

int coscount = 0;

int vowcount = 0;

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

{

if ((str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z'))

{

if (str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u' ||

str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U')

{

vowcount++;

}

else

{

coscount++;

}

}

}

Console.Write("The number of Vowels: {0}\nThe number of Consonants: {1}", vowcount, coscount);

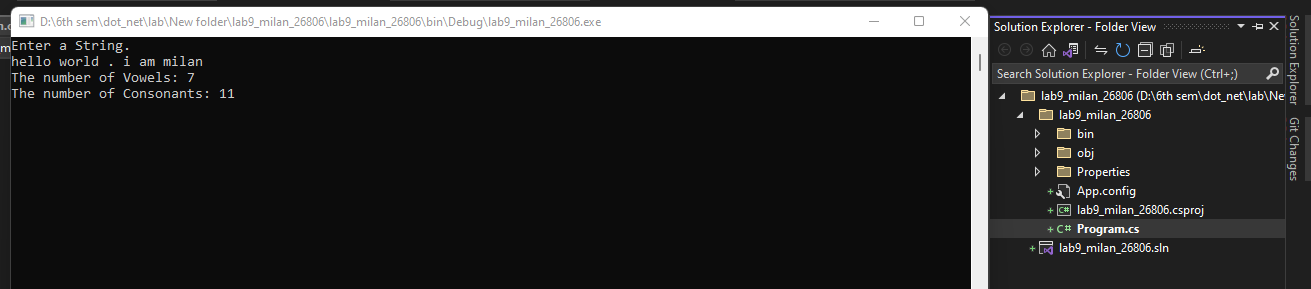
Console.ReadLine();

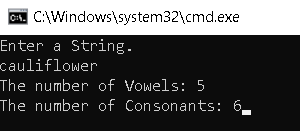
}

}

}

Output:





# Lab: 10

Title: Create student class with properties for id, name, gender and address. It then creates a list<Student> to store instances of this class. The program adds 10 student tot he list, prints the list, and then searches for a student by their address using the FindStudent By Address function.

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab10\_milan\_26806

{

internal class Program

{

static void Main(string[] args)

{

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

st.Add(new Student() { id = 1, Name = "Hari maya", Address = "Kathmandu", Gender = "Female" });

st.Add(new Student() { id = 2, Name = "Dhan maya", Address = "Lalitpur", Gender = "Female" });

st.Add(new Student() { id = 3, Name = "Gun Maya", Address = "Kathmandu", Gender = "Female" });

st.Add(new Student() { id = 4, Name = "Man maya", Address = "Lalitpur", Gender = "Female" });

st.Add(new Student() { id = 5, Name = "sun Maya", Address = "Kathmandu", Gender = "Female" });

st.Add(new Student() { id = 6, Name = "Hari kumar", Address = "Bhaktapur", Gender = "Male" });

st.Add(new Student() { id = 7, Name = "Dhan kumar", Address = "Kathmandu", Gender = "Male" });

st.Add(new Student() { id = 9, Name = "Sun kumar", Address = "Kathmandu", Gender = "Male" });

st.Add(new Student() { id = 10, Name = "Gun Kumar", Address = "Bhaktapur", Gender = "Male" });

st.Add(new Student() { id = 10, Name = "Gun Kumar", Address = "Bhaktapur", Gender = "Male" });

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*List of Students\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

foreach (var item in st)

{

Console.WriteLine("Name:{0} Address:{1} Gender:{2}", item.Name, item.Address, item.Gender);

}

List<Student> filterStudent = FindStudentByAddress(st, "Kathmandu");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Students with Address Kathmandu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

foreach (var item in filterStudent)

{

Console.WriteLine("Name:{0} Address:{1} Gender:{2}", item.Name, item.Address, item.Gender);

}

Console.ReadLine();

}

public static List<Student> FindStudentByAddress(List<Student> students, String searchAddress)

{

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

foreach (Student item in students)

{

if (item.Address == searchAddress)

{

filterstudent.Add(item);

}

}

return filterstudent;

}

}

public class Student

{

public int id { get; set; }

public string Name { get; set; }

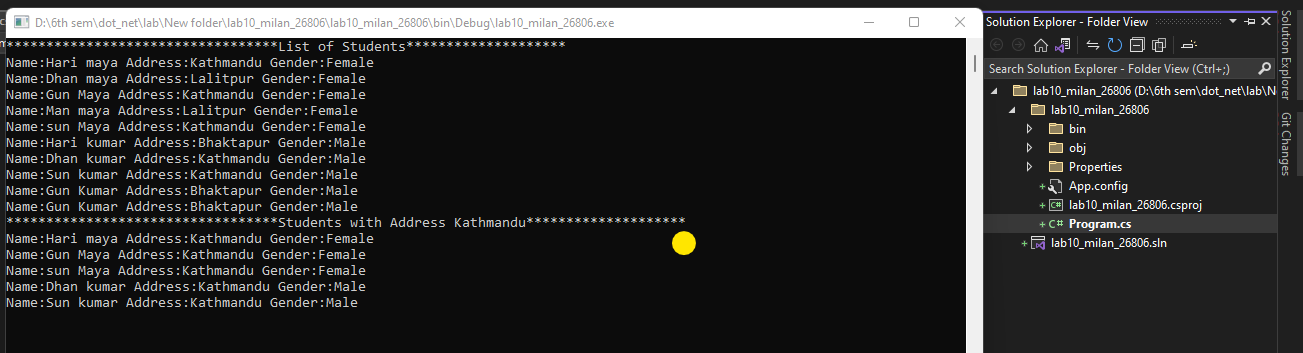
public string Address { get; set; }

public string Gender { get; set; }

}

}

Output:



# Lab11: Write a C# program to call base class constructor using “base” Keyword

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab11\_milan\_26806

{

class framework

{

public string version;

public framework()

{

this.version = "4.7.9";

}

}

class version : framework

{

public string framework;

public version() : base()

{

this.framework = "Dot.net";

}

}

internal class Program

{

static void Main(string[] args)

{

version v = new version();

Console.WriteLine("The Framework is " + v.framework);

Console.WriteLine("The version is " + v.version);

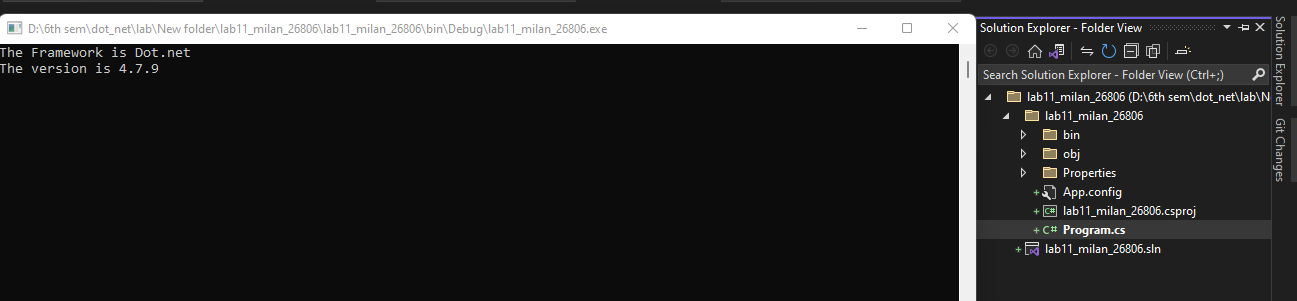
Console.ReadLine();

}

}

}

Output:



# Lab 12: Write a C# program to calculate area and paint cost using inheritance use base keyword to initialize length and breadth.

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab12\_milan\_26806

{

class rectangle

{

public int width;

public int height;

public rectangle(int x, int y)

{

width = x;

height = y;

}

public int getarea()

{

return width \* height;

}

public void display()

{

Console.WriteLine("Length: {0}", height);

Console.WriteLine("Width: {0}", width);

Console.WriteLine("Area: {0}", getarea());

}

}

class tabletop : rectangle

{

public int cost;

public tabletop(int length, int width) : base(length, width) { }

public int getcost(int price)

{

cost = price \* getarea();

return cost;

}

public void display(int price)

{

base.display();

Console.WriteLine("cost: " + getcost(price));

}

}

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter the length");

int a = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the breadth");

int b = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the painting price");

int price = Convert.ToInt32(Console.ReadLine());

tabletop t = new tabletop(a, b);

t.display(price);

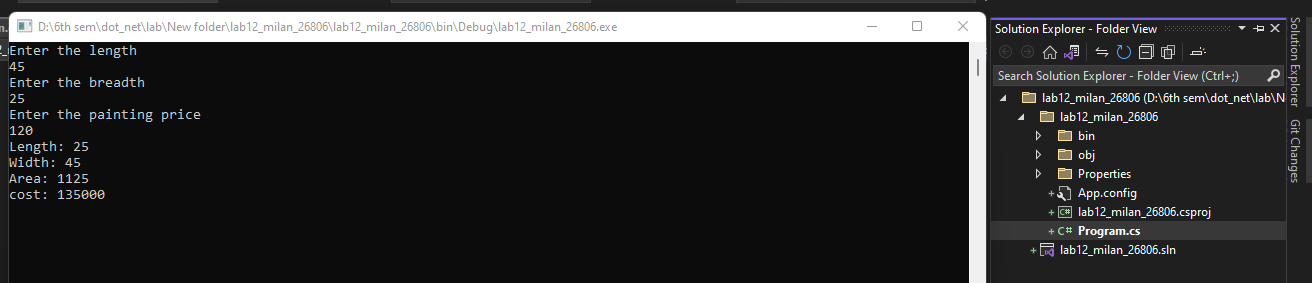
Console.Read();

}

}

}

Output:



# Lab 13: Write a C# program to illustrate hierarchical inheritance with virtual method.

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab13\_milan\_26806

{

class car

{

public virtual string carname()

{

return "Car";

}

}

class Ferrari : car

{

public override string carname()

{

return "Ferrari";

}

}

class Labmorghini : car

{

public override string carname()

{

return "Lamborghini";

}

}

internal class Program

{

static void Main(string[] args)

{

car c1 = new Ferrari();

car c2 = new Labmorghini();

Console.WriteLine(c1.carname());

Console.WriteLine(c2.carname());

Console.ReadLine();

}

}

}

Output:

