# Tutorial\_2

## : Predict and write output for the following code.

using System;

namespace DecisionMaking

{

class Program

{

static void Main(string[] args)

{

/\* local variable deﬁnition \*/ int a = 10;

/\* check the boolean condition using if statement \*/ if (a < 20)

{

/\* if condition is true then print the following \*/ Console.WriteLine("a is less than 20");

}

Console.WriteLine("value of a is : {0}", a); Console.ReadLine();

}

}

}

## CODE:

using System;

namespace DecisionMaking

{

class Program

{

static void Main(string[] args)

{

int a = 10;

if (a < 20)

{

Console.WriteLine("a is less than 20");

}

Console.WriteLine("value of a is : {0}", a);

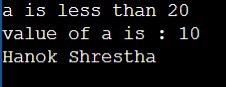
Console.WriteLine("Hanok Shrestha"); Console.ReadLine();

}

}

}

## OUTPUT:



1. **: Write missing statement to get the desired output.**

using System;

namespace DecisionMaking

{

class Program

{

static void Main(string[] args)

{

/\* local variable deﬁnition \*/ int a = 100;

/\* check the boolean condition \*/ if (a < 20)

{

/\* if condition is true then print the following \*/

Console.WriteLine("a is less than 20");

}

else

{

/\* if condition is false then print the following \*/

//… Missing

statement-1… //

}

//… Missing

statement-2… //

Console.ReadLine();

}

}

}

Output

a is not less than 20 value of a is : 100 **CODE:**

using System;

namespace DecisionMaking

{

class Decision

{

static void Main(string[] args)

{

int a = 100;

if (a < 20)

{

Console.WriteLine("a is less than 20");

}

else

{

Console.WriteLine("a is not less than 20");

}

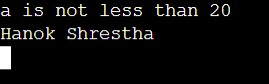
Console.WriteLine("Hanok Shrestha"); Console.ReadLine();

}

}

}

## OUTPUT:



1. **: Correct the following code and write output for the corrected code.**

using System;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

char ﬁrstName = "John"; char lastName = "Doe";

Console.WriteLine("Name: " + ﬁrstName + " " + lastName);

Console.WriteLine("Please enter a new ﬁrst name:") ﬁrstName = Console.ReadLine();

Console.WriteLine("New name: " ﬁrstName " " lastName);

Console.ReadLine();

}

}

}

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_2

{

internal class \_3t

{

static void Main(string[] args)

{

string firstName = "John"; string lastName = "Doe";

Console.WriteLine("Name: " + firstName + " " + lastName);

Console.WriteLine("Please enter a new first name:"); firstName = Console.ReadLine();

Console.WriteLine("New name: " + firstName + " " + lastName); Console.WriteLine("Hanok Shrestha");

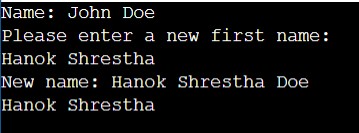
Console.ReadLine();

}

}

}

## OUTPUT:



1. **: Input two number A and B. perform different operations using different operators and different data types available in C#. (Note : Follow all the operators and data types to do above task. Use Online help whenever necessary.)**

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_2

{

internal class \_4t

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter the first number (A): "); string inputA = Console.ReadLine();

double A = Convert.ToDouble(inputA);

Console.Write("Enter the second number (B): "); string inputB = Console.ReadLine();

double B = Convert.ToDouble(inputB);

// Addition

double addition = A + B; Console.WriteLine($"Addition (A + B): {addition}");

// Subtraction

double subtraction = A - B; Console.WriteLine($"Subtraction (A - B): {subtraction}");

// Multiplication

double multiplication = A \* B; Console.WriteLine($"Multiplication (A \* B): {multiplication}");

// Division if (B != 0)

{

double division = A / B; Console.WriteLine($"Division (A / B): {division}");

}

else

{

Console.WriteLine("Cannot divide by zero.");

}

// Modulus (remainder) if (B != 0)

{

double modulus = A % B; Console.WriteLine($"Modulus (A % B): {modulus}");

}

else

{

Console.WriteLine("Cannot perform modulus with zero.");

}

// Increment

double incrementA = A++; double incrementB = ++B;

Console.WriteLine($"Increment A: {incrementA}, Increment B: {incrementB}");

// Decrement

double decrementA = A--; double decrementB = --B;

Console.WriteLine($"Decrement A: {decrementA}, Decrement B: {decrementB}");

// Bitwise AND int intA = (int)A; int intB = (int)B;

int bitwiseAND = intA & intB;

Console.WriteLine($"Bitwise AND (A & B): {bitwiseAND}");

// Bitwise OR

int bitwiseOR = intA | intB; Console.WriteLine($"Bitwise OR (A | B): {bitwiseOR}");

// Bitwise XOR

int bitwiseXOR = intA ^ intB;

Console.WriteLine($"Bitwise XOR (A ^ B): {bitwiseXOR}");

// Bitwise NOT

int bitwiseNOT\_A = ~intA; int bitwiseNOT\_B = ~intB;

Console.WriteLine($"Bitwise NOT (~A): {bitwiseNOT\_A}, Bitwise NOT (~B): {bitwiseNOT\_B}");

// Logical AND

bool logicalAND = (A > 0) && (B > 0);

Console.WriteLine($"Logical AND (A > 0) && (B > 0): {logicalAND}");

// Logical OR

bool logicalOR = (A > 0) || (B > 0);

Console.WriteLine($"Logical OR (A > 0) || (B > 0): {logicalOR}");

// Logical NOT

bool logicalNOT\_A = !(A > 0); bool logicalNOT\_B = !(B > 0);

Console.WriteLine($"Logical NOT !(A > 0): {logicalNOT\_A}, Logical NOT !(B > 0): {logicalNOT\_B}");

// Relational Operators

Console.WriteLine($"A == B: {A == B}, A != B: {A != B}, A > B: {A > B}, A < B: {A < B}, A >= B: {A >= B}, A <= B: {A <= B}");

// Ternary Operator

double max = A > B ? A : B; Console.WriteLine($"Max (A > B ? A : B): {max}");

// Casting

int castedA = (int)A;

double castedB = (double)B;

Console.WriteLine($"Casted A (int): {castedA}, Casted B (double): {castedB}");

// Adding the line to print "Hanok Shrestha" Console.WriteLine("Hanok Shrestha");

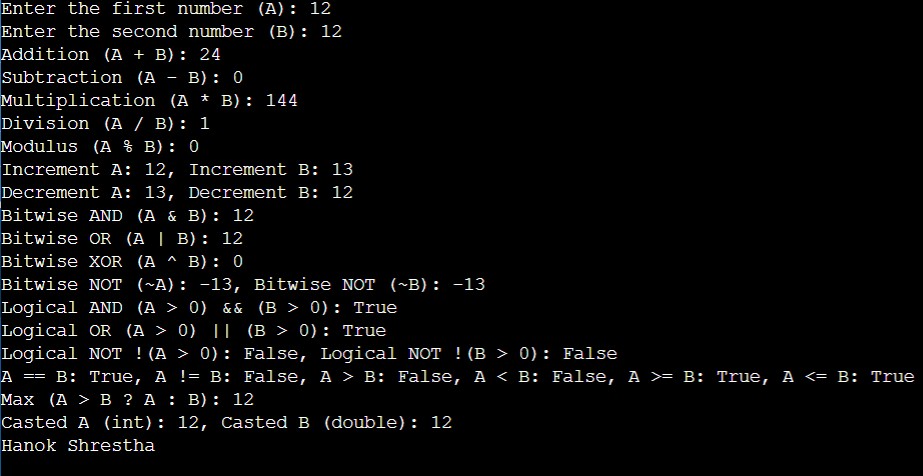
}

}

}

}

## OUTPUT:



1. **: Rearrange the given code to correct the program. The resultant program will be to enter 5 elements into an array and print sum of these elements.**

using System;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

for (int i = 0; i < 5; i++)

{

string str = Console.ReadLine();

}

for (int i = 0; i < 5; i++)

{

sum = sum + arr[i];

}

Console.WriteLine("Sum of Elements : {0}",sum); int[] arr = new int[5];

int sum = 0;

arr[i] = Convert.ToInt32(str); Console.Write("Enter Element {0}: ", i); Console.Read();

}

}

}

Output:

Enter Element 0: 1

Enter Element 1: 2

Enter Element 2: 3

Enter Element 3: 4

Enter Element 4: 5 Sum of Elements : 15 **CODE:**

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_2

{

internal class \_5t

{

static void Main(string[] args)

{

int[] arr = new int[5]; int sum = 0;

for (int i = 0; i < 5; i++)

{

Console.Write("Enter Element {0}: ", i); string str = Console.ReadLine();

arr[i] = Convert.ToInt32(str);

}

for (int i = 0; i < 5; i++)

{

sum = sum + arr[i];

}

Console.WriteLine("Sum of Elements : {0}", sum);

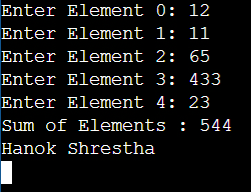
// Adding the line to print "Hanok Shrestha" Console.WriteLine("Hanok Shrestha"); Console.ReadLine();

}

}

}

## OUTPUT:



**6: Write missing statement to get the desired output.**

using System; public class Hello3

{

public static void Main(string[] args)

{

Console.WriteLine("Hello, World!");

Console.WriteLine("You entered the following {0} command line arguments:", args.Length );

//… Missing

statement-1… //

//… Missing

statement-2… //

//… Missing

statement-3… //

//… Missing

statement-4… //

}

}

Output:

Hello, World!

You entered the following 4 command line arguments: A

B C D

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_2

{

internal class \_6t

{

public static void Main(string[] args)

{

Console.WriteLine("Hello, World!");

Console.WriteLine("You entered the following {0} command line arguments:", args.Length);

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

{

Console.WriteLine("Argument {0}: {1}", i, args[i]);

}

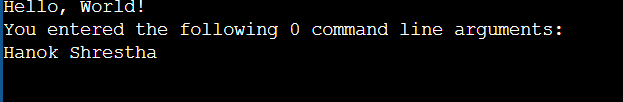
// Adding the line to print "Hanok Shrestha" Console.WriteLine("Hanok Shrestha"); Console.ReadLine();

}

}

}

## OUTPUT:



1. **: Predict and write the output of the given code.**

using System;

namespace CalculatorApplication

{

class NumberManipulator

{

public void swap(ref int x, ref int y)

{

int temp;

temp = x; /\* save the value of x \*/ x = y; /\* put y into x \*/

y = temp; /\* put temp into y \*/

}

}

class TestRef

{

static void Main(string[] args)

{

NumberManipulator n = new NumberManipulator();

/\* local variable deﬁnition \*/ int a = 100;

int b = 200;

Console.WriteLine("Before swap, value of a : {0}", a); Console.WriteLine("Before swap, value of b : {0}", b);

/\* calling a function to swap the values \*/ n.swap(ref a, ref b);

Console.WriteLine("After swap, value of a : {0}", a); Console.WriteLine("After swap, value of b : {0}", b);

Console.ReadLine();

}

}

}

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_2

{

internal class \_7t

{

class NumberManipulator

{

public void swap(ref int x, ref int y)

{

int temp;

temp = x; /\* save the value of x \*/ x = y; /\* put y into x \*/

y = temp; /\* put temp into y \*/

}

}

class TestRef

{

static void Main(string[] args)

{

NumberManipulator n = new NumberManipulator();

/\* local variable definition \*/ int a = 100;

int b = 200;

Console.WriteLine("Before swap, value of a : {0}", a); Console.WriteLine("Before swap, value of b : {0}", b);

/\* calling a function to swap the values \*/ n.swap(ref a, ref b);

Console.WriteLine("After swap, value of a : {0}", a); Console.WriteLine("After swap, value of b : {0}", b);

Console.ReadLine();

// Adding the line to print "Hanok Shrestha" Console.WriteLine("Hanok Shrestha");

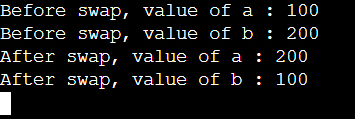
}

}

}

}

## OUTPUT:



1. **: Find out error code and correct it. Write the output of the corrected code.**

using System;

namespace CalculatorApplication

{

class NumberManipulator

{

public int getValues(out int x, out int y, out int z )

{

Console.WriteLine("Enter the ﬁrst value: "); x = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the second value: "); y = Convert.ToInt32(Console.ReadLine());

sum = “x” + “y” + “z”; return “sum”;

}

}

class TestOut

{

static void Main(string[] args)

{

NumberManipulator n = new NumberManipulator();

/\* local variable deﬁnition \*/ int a , b, c, sum;

/\* calling a function to get the values \*/ sum = n.getValues(out a, out b, out c);

Console.WriteLine("After method call, value of a : {0}", a); Console.WriteLine("After method call, value of b : {0}", b); Console.WriteLine("After method call, value of c : {0}", c); Console.WriteLine("Sum : {0}", );

}

}

}

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_2

{

internal class \_8t

{

class NumberManipulator

{

public int getValues(out int x, out int y, out int z)

{

Console.WriteLine("Enter the first value: "); x = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the second value: "); y = Convert.ToInt32(Console.ReadLine());

z = 0; // Initialize z to 0 int sum = x + y + z; return sum;

}

}

class TestOut

{

static void Main(string[] args)

{

NumberManipulator n = new NumberManipulator();

/\* local variable definition \*/ int a, b, c, sum;

/\* calling a function to get the values \*/ sum = n.getValues(out a, out b, out c);

Console.WriteLine("After method call, value of a : {0}", a); Console.WriteLine("After method call, value of b : {0}", b); Console.WriteLine("After method call, value of c : {0}", c); Console.WriteLine("Sum : {0}", sum);

// Adding the line to print "Hanok Shrestha" Console.WriteLine("Hanok Shrestha"); Console.ReadLine();

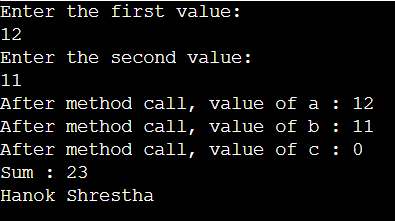
}

}

}

}

## OUTPUT:



1. **: Given an array A containing 2\*N+2 positive numbers, out of which 2\*N numbers exist in pairs whereas the other two number occur exactly once and are distinct. Find the other two numbers.**

## Example 1:Input:

N = 2

arr[] = {1, 2, 3, 2, 1, 4}

## Output:

3 4

## Explanation:

3 and 4 occur exactly once.

## Example 2:

**Input:**

N = 1

arr[] = {2, 1, 3, 2}

## Output:

1 3

## Explanation:

1 3 occur exactly once.

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_2

{

internal class \_9t

{

class Program

{

static void FindUniqueNumbers(int[] arr, int n)

{

int xorResult = 0;

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

{

xorResult ^= arr[i];

}

// Finding the rightmost set bit in the xorResult

int rightmostSetBit = xorResult & ~(xorResult - 1);

int num1 = 0, num2 = 0; for (int i = 0; i < n; i++)

{

if ((arr[i] & rightmostSetBit) != 0)

{

num1 ^= arr[i];

}

else

{

num2 ^= arr[i];

}

}

Console.WriteLine(num1 + " " + num2);

}

static void Main(string[] args)

{

int N1 = 2;

int[] arr1 = { 1, 2, 3, 2, 1, 4 };

Console.WriteLine("Output for Example 1:"); FindUniqueNumbers(arr1, N1);

int N2 = 1;

int[] arr2 = { 2, 1, 3, 2 }; Console.WriteLine("Output for Example 2:"); FindUniqueNumbers(arr2, N2);

// Adding the line to print "Hanok Shrestha" Console.WriteLine("Hanok Shrestha"); Console.ReadLine();

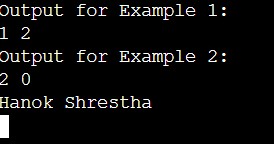
}

}

}

}

## OUTPUT:



**10: Given a matrix mat[][] of size N x M, where every row and column is sorted in increasing order, and a number X is given. The task is to ﬁnd whether element X is present in the matrix or not.**

## Example 1: Input:

N = 3, M = 3

mat[][] = 3 30 38

44 52 54

57 60 69

X = 62

## Output:

0

## Explanation:

62 is not present in the

matrix, so output is 0

## Example 2:

**Input**:

N = 1, M = 6

mat[][] = 18 21 27 38 55 67

X = 55

## Output:

1

## Explanation:

55 is present in the matrix at 5th cell.

## Your Task:

You don't need to read input or print anything. You just have to complete the function matSearch() which takes a 2D matrix **mat**[][], its dimensions **N** and **M** and integer **X** as inputs and returns 1 if the element **X** is present in the matrix and 0 otherwise.

**Expected Time Complexity**: O(N+M).

**Expected Auxiliary Space**: O(1).

## Constraints:

1 <= N, M <= 1005

1 <= mat[][] <= 10000000

1<= X <= 10000000

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_2

{

internal class \_10t

{

class MatrixSearch

{

static int matSearch(int[,] mat, int N, int M, int X)

{

int i = 0, j = M - 1; // Start from the top-right corner of the matrix

while (i < N && j >= 0)

{

if (mat[i, j] == X) // If the element is found

{

return 1;

}

else if (mat[i, j] > X) // If the current element is greater than X, move left

{

j--;

}

else // If the current element is smaller than X, move down

{

i++;

}

}

return 0; // If the element is not found, return 0

}

static void Main()

{

int N1 = 3, M1 = 3;

int[,] mat1 = { { 3, 30, 38 }, { 44, 52, 54 }, { 57, 60, 69 } };

int X1 = 62;

Console.WriteLine("Output for Example 1: " + matSearch(mat1, N1, M1, X1));

int N2 = 1, M2 = 6;

int[,] mat2 = { { 18, 21, 27, 38, 55, 67 } };

int X2 = 55;

Console.WriteLine("Output for Example 2: " + matSearch(mat2, N2, M2, X2));

// Adding the line to print "Hanok Shrestha" Console.WriteLine("Hanok Shrestha");

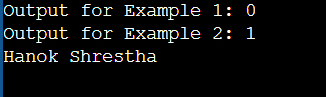
}

}

}

}

## OUTPUT:



**11: Write a program to ﬁnd the sum of N elements of an Array. CODE:**

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks; namespace Tutorial\_2

{

internal class \_11t

{

class Program

{

static void Main(string[] args)

{

int N = 5;

int[] arr = new int[N]; int sum = 0;

// Taking input from the user for the array elements for (int i = 0; i < N; i++)

{

Console.Write("Enter Element {0}: ", i);

arr[i] = Convert.ToInt32(Console.ReadLine());

}

// Calculating the sum of the array elements for (int i = 0; i < N; i++)

{

sum += arr[i];

}

Console.WriteLine("Sum of Elements : {0}", sum); Console.ReadLine();

// Adding the line to print "Hanok Shrestha" Console.WriteLine("Hanok Shrestha"); Console.ReadLine();

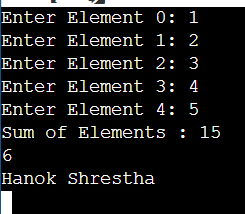
}

}

}

}

## OUTPUT:



**12: Write a program to ﬁnd the element from an Array and print 1 if element is found else print 0.**

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_2

{

internal class \_12t

{

class Program

{

static void Main(string[] args)

{

int N = 5;

int[] arr = new int[N];

// Taking input from the user for the array elements for (int i = 0; i < N; i++)

{

Console.Write("Enter Element {0}: ", i);

arr[i] = Convert.ToInt32(Console.ReadLine());

}

// Taking input for the element to search Console.Write("Enter the element to search: ");

int elementToSearch = Convert.ToInt32(Console.ReadLine()); bool found = false;

// Searching for the element in the array for (int i = 0; i < N; i++)

{

if (arr[i] == elementToSearch)

{

found = true; break;

}

}

// Printing the result if (found)

{

Console.WriteLine("1");

}

else

{

Console.WriteLine("0");

}

Console.ReadLine();

// Adding the line to print "Hanok Shrestha" Console.WriteLine("Hanok Shrestha"); Console.ReadLine();

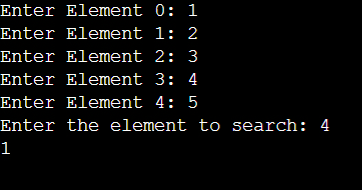
}

}

}

}

## OUTPUT:



1. **Write a Program that will accept the amount and ﬁnd how many minimum no of notes you required for that.**

## (Using the rupee notes of 1, 2, 5, 10, 20, 50, 100, 200, 500, 2000)

**Input: 5748 Output:**

## Notes of Rs.2000 = 2 Notes of Rs.500 = 3 Notes of Rs.200 = 1 Notes of Rs.20 = 2 Notes of Rs.10 = 0 Notes of Rs.5 = 1 Notes of Rs.2 = 1 Notes of Rs.1 = 1

**CODE:**

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_2

{

internal class \_13t

{

class Program

{

static void Main(string[] args)

{

int[] notes = { 2000, 500, 200, 100, 50, 20, 10, 5, 2, 1 };

int[] count = new int[10];

Console.Write("Enter the amount: ");

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

for (int i = 0; i < 10; i++)

{

count[i] = amount / notes[i]; amount %= notes[i];

}

for (int i = 0; i < 10; i++)

{

Console.WriteLine("Notes of Rs.{0} = {1}", notes[i], count[i]);

}

Console.ReadLine();

// Adding the line to print "Hanok Shrestha" Console.WriteLine("Hanok Shrestha"); Console.ReadLine();

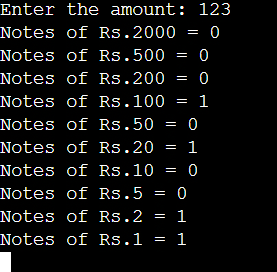
}

}

}

}

## OUTPUT:



1. **Write a Program to ﬁnd the eligibility of admission for a professional course based on the following criteria:**

## Marks in Maths >=65 Marks in Phy >=55 Marks in Chem>=50 and

**Total in all three subject >=180 or Total in Math and Physics >=140 INPUT:**

## Input the marks obtained in Maths :72 Input the marks obtained in Physics :65 Input the marks obtained in Chemistry :51 OUTPUT:

**The candidate is eligible for admission. CODE:**

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_2

{

internal class \_14t

{

class Program

{

static void Main(string[] args)

{

Console.Write("Input the marks obtained in Maths : "); int mathMarks = Convert.ToInt32(Console.ReadLine());

Console.Write("Input the marks obtained in Physics : "); int physicsMarks = Convert.ToInt32(Console.ReadLine());

Console.Write("Input the marks obtained in Chemistry : "); int chemistryMarks = Convert.ToInt32(Console.ReadLine());

int totalMarks = mathMarks + physicsMarks + chemistryMarks; int mathAndPhysicsMarks = mathMarks + physicsMarks;

if (mathMarks >= 65 && physicsMarks >= 55 && chemistryMarks >= 50 && (totalMarks >= 180 || mathAndPhysicsMarks >= 140))

{

Console.WriteLine("The candidate is eligible for admission.");

}

else

{

Console.WriteLine("The candidate is not eligible for admission.");

}

Console.ReadLine();

// Adding the line to print "Hanok Shrestha" Console.WriteLine("Hanok Shrestha"); Console.ReadLine();

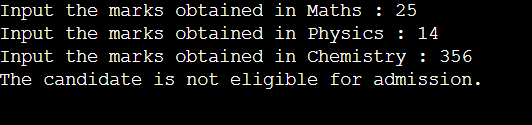
}

}

}

}

## OUTPUT:



1. **Write a Program which accepts name from the user and prints the same INPUT : R K University**

## OUTPUT: R K University CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_2

{

internal class \_15t

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter your name: "); string name = Console.ReadLine(); Console.WriteLine("Name: " + name);

Console.ReadLine();

// Adding the line to print "Hanok Shrestha" Console.WriteLine("Hanok Shrestha"); Console.ReadLine();

}

}

}

}

## OUTPUT:



**Tutorial\_3**

## : Draw a real picture for class and object. Differentiate class and object in terms of diagram only.

**Perform following tasks.**

Task 1: Create a class

Task 2: Add few data members as private

Task 3: Add few methods as public to work on defined data members Task 4: Create a Demo class with main method.

Task 5: Create at least two objects of a class defined in Task 1 into main method and call all methods using that object.

Task 6: Write comment for each important portion of code like data members’ declaration, methods, some important logic etc.

Task 7: Summarize above solution in your own few words to visualize the solution to the end user.

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_3

{

internal class q1

{

// Task 1: Create a class class MyClass

{

// Task 2: Add few data members as private private int privateData1;

private string privateData2;

// Task 3: Add few methods as public to work on defined data members public MyClass()

{

privateData1 = 0; privateData2 = "default";

}

public int GetPrivateData1()

{

return privateData1;

}

public void SetPrivateData1(int value)

{

privateData1 = value;

}

public string GetPrivateData2()

{

return privateData2;

}

public void SetPrivateData2(string value)

{

privateData2 = value;

}

}

// Task 4: Create a Demo class with the main method. class Demo

{

// Task 5: Create at least two objects of a class defined in Task 1 into the main method and call all methods using that object.

static void Main(string[] args)

{

// Create objects of MyClass MyClass obj1 = new MyClass(); MyClass obj2 = new MyClass();

// Call methods on the objects obj1.SetPrivateData1(40); obj1.SetPrivateData2("Hello"); obj2.SetPrivateData1(12); obj2.SetPrivateData2("Ramesh");

// Display data using the Get methods Console.WriteLine("Object 1 data1: " + obj1.GetPrivateData1()); Console.WriteLine("Object 1 data2: " + obj1.GetPrivateData2()); Console.WriteLine("Object 2 data1: " + obj2.GetPrivateData1()); Console.WriteLine("Object 2 data2: " + obj2.GetPrivateData2()); Console.WriteLine("Hanok Shrestha");

Console.ReadLine();

}

}

}

}

// Task 7: Summarize above solution in your own few words to visualize the solution to the end-user.

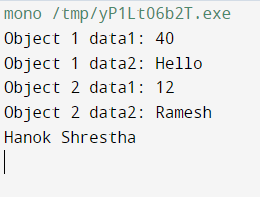
/\*

The C# program defines a class "MyClass" with private data members and public methods to work with them. A "Demo" class showcases how to create objects of "MyClass" and use the methods to set and retrieve data. The output demonstrates the successful interaction between objects and methods in C#.

\*/

// Note: Don't forget to add "using System;" at the beginning of the code for the Console.WriteLine() to work properly.

## OUTPUT:



1. **: Define a class Clock with three *private* integer data members hour, min and sec. Define a no argument constructor to initialize time value to 12:00:00. Define a three argument constructor to initialize the time.**

## Define a methods to

* + **Increment time to next second.**

## Display the time value.

* + **Return the hour (*int getHour()*)**
  + **Return the minute (*int getMinute()*)**
  + **Return the seconds (*int getSeconds()*)**

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_3

{

internal class q2

{

class Clock

{

// Private data members private int hour;

private int min; private int sec;

// No argument constructor to initialize time value to 12:00:00 public Clock()

{

hour = 12;

min = 0;

sec = 0;

// Three argument constructor to initialize the time public Clock(int hour, int min, int sec)

{

this.hour = hour; this.min = min; this.sec = sec;

}

// Method to increment time to next second public void IncrementTime()

{

sec++;

if (sec >= 60)

{

sec = 0; min++;

if (min >= 60)

{

min = 0; hour++;

if (hour >= 24)

{

hour = 0;

}

}

}

}

// Method to display the time value public void DisplayTime()

{

string formattedHour = hour.ToString("D2"); string formattedMin = min.ToString("D2"); string formattedSec = sec.ToString("D2");

Console.WriteLine($"Time: {formattedHour}:{formattedMin}:{formattedSec}");

}

// Method to return the hour public int GetHour()

{

return hour;

}

// Method to return the minute public int GetMinute()

{

return min;

}

// Method to return the seconds public int GetSeconds()

{

return sec;

}

static void Main(string[] args)

// Create a Clock object using the no-argument constructor Clock clock1 = new Clock();

// Display the initial time Console.WriteLine("Initial Time:"); clock1.DisplayTime();

// Increment the time to the next second clock1.IncrementTime();

// Display the updated time Console.WriteLine("Updated Time:"); clock1.DisplayTime();

Console.WriteLine("\n Hanok Shrestha"); Console.ReadLine();

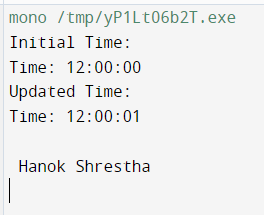
}

}

}

}

## OUTPUT:



1. **: Define a Student class with appropriate data members, property, constructors, methos etc. Define another class called TestStudent within the same .cs file. Also create an object of student class and demonstrate the use of student class.**

## CODE:

using System;

using System.Collections.Generic;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_3

{

internal class q3

{

class Student

{

// Data members private string name; private int age;

private string rollNumber;

// Properties

public string Name

{

get { return name; } set { name = value; }

}

public int Age

{

get { return age; } set { age = value; }

}

public string RollNumber

{

get { return rollNumber; } set { rollNumber = value; }

}

// Constructors public Student()

{

name = "Prakash Khatri"; age = 18;

rollNumber = "12";

}

public Student(string name, int age, string rollNumber)

{

this.name = name; this.age = age;

this.rollNumber = rollNumber;

}

// Method

public void DisplayInfo()

{

Console.WriteLine("Name: " + name); Console.WriteLine("Age: " + age); Console.WriteLine("Roll Number: " + rollNumber);

}

}

class TestStudent

static void Main(string[] args)

{

// Create an object of Student class Student student1 = new Student(); student1.DisplayInfo();

Console.WriteLine();

// Create another object of Student class with custom values Student student2 = new Student("Hari", 20, "67"); student2.DisplayInfo();

Console.WriteLine("\n Hanok Shrestha"); Console.ReadLine();

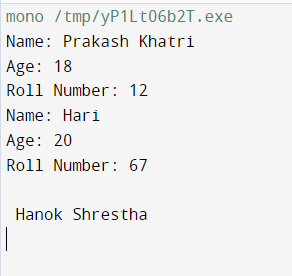
}

}

}

}

## OUTPUT:



1. **: Use above program classes and create objects for 5 students and demonstrate the use student class.**

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_3

{

{

class Student

{

// Data members private string name; private int age;

private string rollNumber;

// Properties

public string Name

{

get { return name; } set { name = value; }

}

public int Age

{

get { return age; } set { age = value; }

}

public string RollNumber

{

get { return rollNumber; } set { rollNumber = value; }

}

// Constructors public Student()

{

name = "Ramesh Pokhrel"; age = 27;

rollNumber = "61";

}

public Student(string name, int age, string rollNumber)

{

this.name = name; this.age = age;

this.rollNumber = rollNumber;

}

// Method

public void DisplayInfo()

{

Console.WriteLine("Name: " + name); Console.WriteLine("Age: " + age); Console.WriteLine("Roll Number: " + rollNumber); Console.WriteLine();

}

}

class TestStudent

{

static void Main(string[] args)

{

// Create objects of Student class for 5 students

Student student1 = new Student("Ram", 22, "65"); Student student2 = new Student("Hari", 22, "53"); Student student3 = new Student("Giri", 24, "66"); Student student4 = new Student("Suresh", 25, "55"); Student student5 = new Student("John", 20, "54");

// Demonstrate the use of Student class for each student Console.WriteLine("Student 1 Information:"); student1.DisplayInfo();

Console.WriteLine("Student 2 Information:"); student2.DisplayInfo();

Console.WriteLine("Student 3 Information:"); student3.DisplayInfo();

Console.WriteLine("Student 4 Information:"); student4.DisplayInfo();

Console.WriteLine("Student 5 Information:"); student5.DisplayInfo();

Console.WriteLine("\n Hanok Shhrestha"); Console.ReadLine();

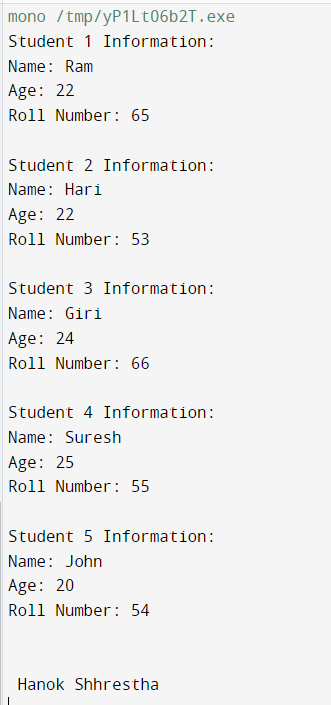
}

}

}

}

## OUTPUT:



1. **: Rearrange the given code to get the desired output.**

using System;

namespace ConsoleApplication

{

class Product

{

public Product(int pcd, String pnm, String mnm)

{

mname = mnm;

}

public void Display()

{

Console.WriteLine("\nManufacturer Name:= " + mname);

}

}

int pcode;

String pname, mname; Console.WriteLine("\nProduct Code:= " + pcode); Console.WriteLine("\nProduct Name:= " + pname); pcode = pcd;

pname = pnm;

public class TestProduct

{

public static void Main(string[] args)

{

int n = args.Length;

if (n < 3)

{

Console.WriteLine("Syntax Error\n"); Console.WriteLine("Must Have THREE Arguments\n");

Console.WriteLine("Please, Write as [csc TestProduct ProductCode ProductName Manufacturer] \n");

}

else

{

Product p=new Product(pcd,pnm,mnm); p.Display();

Console.Read();

int pcd = Convert.ToInt32(args[0]); String pnm = args[1];

String mnm = args[2];

}

}

}

}

Output:

Product Code:= P001 Product Name:= Mouse

Manufacturer Name:= Logitech

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_3

{

internal class q5

{

class Product

{

String pcode;

String pname, mname;

public Product(String pcd, String pnm, String mnm)

{

pcode = pcd; pname = pnm; mname = mnm;

}

public void Display()

{

Console.WriteLine("\nProduct Code:= " + pcode); Console.WriteLine("\nProduct Name:= " + pname); Console.WriteLine("\nManufacturer Name:= " + mname);

}

}

public class TestProduct

{

public static void Main(string[] args)

{

int n = args.Length;

if (n < 3)

{

Console.WriteLine("Syntax Error\n"); Console.WriteLine("Must Have THREE Arguments\n");

Console.WriteLine("Please, Write as [csc TestProduct ProductCode ProductName Manufacturer] \n");

}

else

{

string pcd = args[0]; String pnm = args[1]; String mnm = args[2];

Product p = new Product(pcd, pnm, mnm); p.Display();

Console.Read();

}

Console.WriteLine("\n Hanok Shrestha"); Console.ReadLine();

}

}

}

}

## OUTPUT:



1. **: Complete the following code that will generate the given output: Solution:**

using System;

namespace LineApplication

{

class Line

{

private double length; // Length of a line public Line()

{

//………………………………Missing statement-1 //

//………………………………Missing statement-2 //

}

public void setLength( double len )

{

//………………………………Missing statement-3 //

}

public double getLength()

{

//………………………………Missing statement-4 //

}

}

class TestLine

{

static void Main(string[] args)

{

Line line = new Line();

// set line length

Console.WriteLine("Length of line : {0}", line.getLength());

// set line length

Console.WriteLine("Length of line : {0}", line.getLength()); Console.ReadKey();

}

}

}

Output:

Object is being created, length = 10 Length of line : 10

Length of line : 6

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_3

{

internal class q6

{

class Line

{

private double length; // Length of a line

public Line()

{

length = 10; // Missing statement-1: Set the initial length to 10 Console.WriteLine("Object is being created, length = " + length);

}

public void setLength(double len)

{

length = len; // Missing statement-3: Set the length to the given value

}

public double getLength()

{

return length; // Missing statement-4: Return the length of the line

}

}

class TestLine

{

static void Main(string[] args)

{

Line line = new Line();

// set line length

line.setLength(10); // Missing statement-2: Set the line length to 6 Console.WriteLine("Length of line : {0}", line.getLength());

// set line length line.setLength(6);

Console.WriteLine("Length of line : {0}", line.getLength()); Console.ReadKey();

Console.WriteLine("\n Hanok Shrestha"); Console.ReadKey();

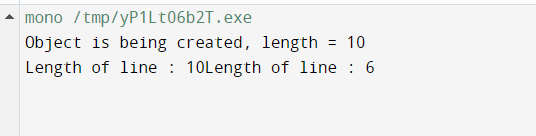
}

}

}

}

## OUTPUT:



1. **: Define EnrolmentNo and Name properties for the Student class and demonstrate use of these properties along with required data members, methods and constructors.**

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_3

{

internal class q7

{

class Student

{

// Data members

private int enrolmentNo; private string name; private int age;

private string rollNumber;

// Properties

public int EnrolmentNo

{

get { return enrolmentNo; } set { enrolmentNo = value; }

}

public string Name

{

get { return name; } set { name = value; }

}

// Constructors public Student()

{

enrolmentNo = 2198603; name = "Bikesh Shrestha"; age = 24;

rollNumber = "67";

}

public Student(int enrolmentNo, string name, int age, string rollNumber)

{

this.enrolmentNo = enrolmentNo; this.name = name;

this.age = age; this.rollNumber = rollNumber;

}

// Method

public void DisplayInfo()

{

Console.WriteLine("Enrolment No: " + enrolmentNo); Console.WriteLine("Name: " + name); Console.WriteLine("Age: " + age); Console.WriteLine("Roll Number: " + rollNumber); Console.WriteLine();

}

}

class TestStudent

{

static void Main()

{

// Create an object of Student class using the default constructor Student student1 = new Student();

student1.DisplayInfo();

// Create another object of Student class using the parameterized constructor Student student2 = new Student(1001, "Janak", 20, "56"); student2.DisplayInfo();

Console.WriteLine("\n Hanok Shrestha"); Console.ReadLine();

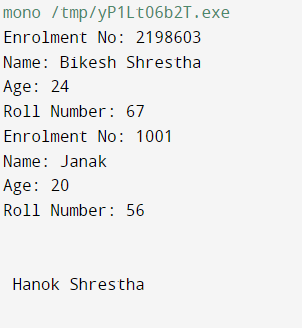
}

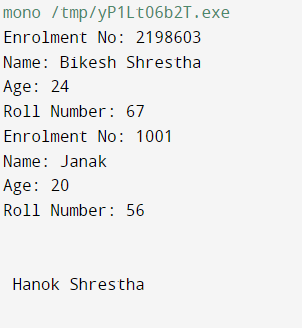
}

}

}

## OUTPUT:





**Tutorial – 1**

1. Write a C program to print “Hello World” on the output screen.

using System; class Program

{

static void Main()

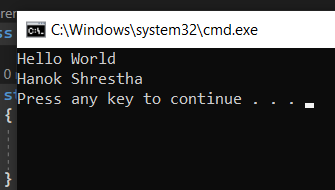
{

Console.WriteLine("Hello World"); Console.WriteLine("Hanok Shrestha");

}

}

**Output:**



1. : Design your profile page as given below.

using System; class Program

{

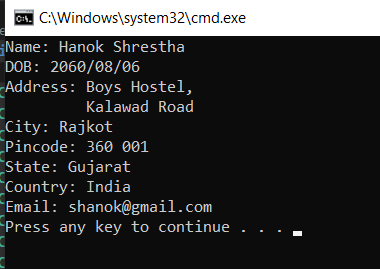
static void Main()

{

Console.WriteLine("Name: Hanok Shrestha"); Console.WriteLine("DOB: 2060/08/06"); Console.WriteLine("Address: Boys Hostel,"); Console.WriteLine(" Kalawad Road"); Console.WriteLine("City: Rajkot"); Console.WriteLine("Pincode: 360 001"); Console.WriteLine("State: Gujarat"); Console.WriteLine("Country: India"); Console.WriteLine("Email: [shanok@gmail.com](mailto:shanok@gmail.com)");

}

}



## : Find out whether the given number is odd or even.

using System; class Program

{

static void Main()

{

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

int number = int.Parse(Console.ReadLine());

if (number % 2 == 0)

{

Console.WriteLine("The number is even.");

}

else

{

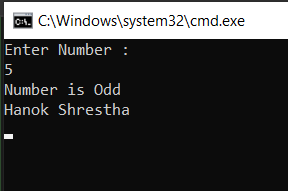
Console.WriteLine("The number is odd.");

}

Console.WriteLine("Hanok Shrestha");

}

}



## : Rearrange the given code to correct the program. The resultant program will be to input a number and print whether the given number is odd or even.

using System;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter Number : "); string str = Console.ReadLine();

int x = Convert.ToInt32(str);

if (x % 2 == 0)

{

Console.WriteLine("Number is Even");

}

else

{

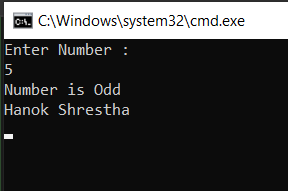
Console.WriteLine("Number is Odd");

}

Console.WriteLine("Hanok Shrestha") Console.Read();

}

}}



## : Write output of the program. Also write a comment for each line for the following code.

using System;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int n, fact = 1;

Console.WriteLine("Enter Number : "); // Prompting the user to enter a number string str = Console.ReadLine(); // Reading the input number as a string

n = Convert.ToInt32(str); // Converting the string input to an integer for (int i = 1; i <= n; i++) // Loop to calculate the factorial

{

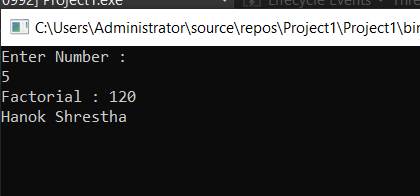
fact = fact \* i; // Calculating the factorial by multiplying the numbers

}

Console.WriteLine("Factorial : {0}", fact); // Outputting the factorial value Console.Read(); // Keeping the console window open

}

}

}

## 7 : Predict and write the output of the given code.

using System;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int a, b, c, result; Console.Write("Enter Number 1: "); string str = Console.ReadLine();

a = Convert.ToInt32(str); Console.Write("Enter Number 2: "); str = Console.ReadLine();

b = Convert.ToInt32(str); Console.Write("Enter Number 3: "); str = Console.ReadLine();

c = Convert.ToInt32(str); result = Sum(a, b, c);

Console.WriteLine("Sum: {0}", result); Console.WriteLine("Hanok Shrestha"); Console.Read();

}

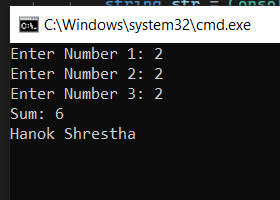
static int Sum(int x, int y, int z)

{

int res;

res = x + y + z; return res;

}

}

}

1. Write a program to convert given name in upper characters. INPUT : John F Kennedy

OUTPUT: JOHN F KENNEDY

using System; class Program

{

static void Main()

{

Console.WriteLine("Enter a name:"); string name = Console.ReadLine();

string upperName = name.ToUpper(); // Convert the name to uppercase

Console.WriteLine("Output: " + upperName + " HANOK SHRESTHA");

}

}

1. Write a Program to convert given name in toggle case. INPUT : JoHn F kEnNedy

OUTPUT: jOhN f KeNneDY using System;

class Program

{

static void Main()

{

Console.WriteLine("Enter a name:"); string name = Console.ReadLine();

string toggleName = ToggleCase(name);

Console.WriteLine("Output: " + toggleName);

}

static string ToggleCase(string str)

{

char[] chars = str.ToCharArray();

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

{

if (char.IsLower(chars[i]))

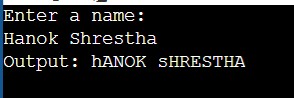
chars[i] = char.ToUpper(chars[i]); else if (char.IsUpper(chars[i]))

chars[i] = char.ToLower(chars[i]);

}

return new string(chars);

}

}

1. Write a Program which accepts mobile no as a string from the user and converts the last 5 digits into X.

INPUT : 1234567890 OUTPUT: 12345XXXXX

using System; class Program

{

static void Main()

{

Console.WriteLine("Enter a mobile number:"); string mobileNumber = Console.ReadLine();

string convertedNumber = ConvertLastDigitsToX(mobileNumber); Console.WriteLine("Output: " + convertedNumber);

}

static string ConvertLastDigitsToX(string number)

{

if (number.Length <= 5)

return new string('X', number.Length); // If the number length is 5 or less, convert all digits to 'X'

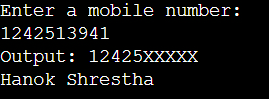
int startIndex = number.Length - 5;

string convertedDigits = new string('X', 5);

return number.Substring(0, startIndex) + convertedDigits;

}

}



1. Write a Program which accepts name and gender from the user. Here, gender may have only 1 character, M or F.

Based on the gender preﬁx the name Mr. & Ms. NAME : Hillary Clinton

GENDER : F

using System; class Program

{

static void Main()

{

Console.WriteLine("Enter your name:"); string name = Console.ReadLine();

Console.WriteLine("Enter your gender (M/F):");

char gender = char.ToUpper(Console.ReadKey().KeyChar);

string preﬁxedName = PreﬁxNameWithGender(name, gender);

Console.WriteLine("\nOutput: " + preﬁxedName);

}

static string PreﬁxNameWithGender(string name, char gender)

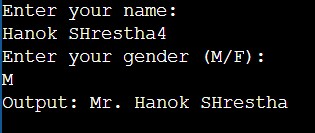
{

if (gender == 'M') return "Mr. " + name;

else if (gender == 'F') return "Ms. " + name;

else

return name;

}

}

1. Write a Program which accepts name from the user and prints the same INPUT : Winston Churchill

OUTPUT: Winston Churchill

using System; class Program

{

static void Main()

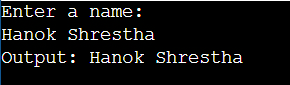
{

Console.WriteLine("Enter a name:"); string name = Console.ReadLine();

Console.WriteLine("Output: " + name);

}

}



1. Write a Program to prints the following series 0 1 1 2 3 5 8 13 21 34 55

using System; class Program

{

static void Main()

{

int n1 = 0; int n2 = 1;

Console.Write(n1 + " " + n2 + " "); // Print the ﬁrst two numbers in the series

for (int i = 0; i < 9; i++) // Loop to calculate and print the next 9 numbers in the series

{

int nextNumber = n1 + n2; Console.Write(nextNumber + " ");

n1 = n2;

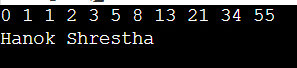
n2 = nextNumber;

}

Console.WriteLine(); // Print a newline to end the series

}

}



1. Write a Program which accepts no from the user and print the same in words. INPUT : 98732

OUTPUT: Nine Eight Seven Three Two using System;

class Program

{

static void Main()

{

Console.WriteLine("Enter a number:"); string number = Console.ReadLine();

string numberInWords = ConvertNumberToWords(number);

Console.WriteLine("Output: " + numberInWords); Console.WriteLine("Hanok Shrestha");

}

static string ConvertNumberToWords(string number)

{

string[] words = { "Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine" };

string numberInWords = ""; foreach (char c in number)

{

if (char.IsDigit(c))

{

int digit = int.Parse(c.ToString()); numberInWords += words[digit] + " ";

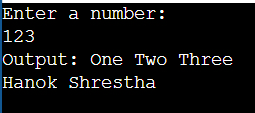
}

}

return numberInWords.TrimEnd();

}

}



## Write a Program to check whether the given no is Armstrong no or not.

using System; class Program

{ static void Main()

{

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

int number = int.Parse(Console.ReadLine()); if (IsArmstrongNumber(number))

{ Console.WriteLine("The number is an Armstrong number."); } else {

Console.WriteLine("The number is not an Armstrong number.");

}

Console.WriteLine("Hanok Shrestha");

}

static bool IsArmstrongNumber(int number)

{

int originalNumber = number;

int result = 0;

int power = GetNumberLength(number); while (number > 0)

{

int digit = number % 10;

result += (int)Math.Pow(digit, power); number /= 10;

}

return (result == originalNumber);

}

static int GetNumberLength(int number)

{

int length = 0;

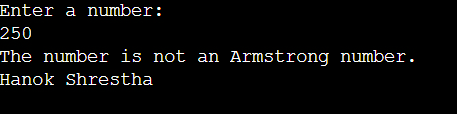
while (number > 0)

{

number /= 10; length++;

}

return length; }}



## Write a program to display a pattern like a right angle triangle using an asterisk The pattern like :

using System;

class Program

{

static void Main()

{

int rows;

Console.WriteLine("Enter the number of rows:"); rows = int.Parse(Console.ReadLine());

for (int i = 1; i <= rows; i++)

{

for (int j = 1; j <= i; j++)

{

Console.Write("\*");

}

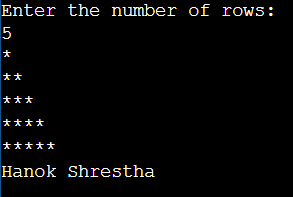
Console.WriteLine("");

}

Console.WriteLine("Hanok Shrestha");

}

}



17. Write a Program to generate following output.

using System; class Program

{

static void Main()

{

int rows;

Console.WriteLine("Enter the number of rows:"); rows = int.Parse(Console.ReadLine());

for (int i = 1; i <= rows; i++)

{

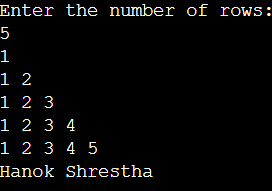
for (int j = 1; j <= i; j++)

{

Console.Write(j + " ");

}

Console.WriteLine();

}

Console.WriteLine("Hanok Shrestha");

}

}

18 Write a program to make such a pattern like a right angle triangle with the number increased by 1.

using System;

class Program

{

static void Main()

{

int rows;

int number = 1;

Console.WriteLine("Enter the number of rows:"); rows = int.Parse(Console.ReadLine());

for (int i = 1; i <= rows; i++)

{

for (int j = 1; j <= i; j++)

{

Console.Write(number + " "); number++;

}

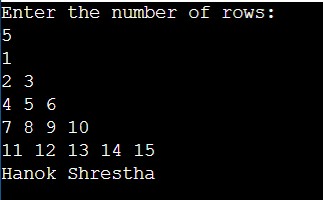
Console.WriteLine();

}

Console.WriteLine("Hanok Shrestha");

}

}



## Write a program to make such a pattern as a pyramid with an asterisk.

using System; class Program

{

static void Main()

{

int rows;

Console.WriteLine("Enter the number of rows:"); rows = int.Parse(Console.ReadLine());

for (int i = 1; i <= rows; i++)

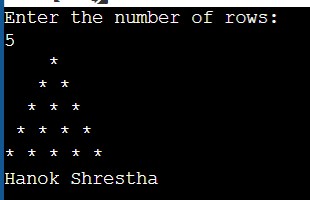
{

for (int j = 1; j <= rows - i; j++)

{Console.Write(" ");}

for (int k = 1; k <= i; k++)

{ Console.Write("\* "); } Console.WriteLine();

}

Console.WriteLine("Hanok Shrestha");

}

}

1. Write a program to make a pyramid pattern with numbers increased by 1. using System;

class Program

{

static void Main()

{

int rows;

int number = 1;

Console.WriteLine("Enter the number of rows:"); rows = int.Parse(Console.ReadLine());

for (int i = 1; i <= rows; i++)

{

for (int j = 1; j <= rows - i; j++)

{

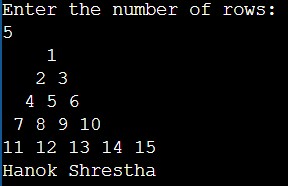
Console.Write(" ");

}

for (int k = 1; k <= i; k++)

{

Console.Write(number + " "); number++;

}

Console.WriteLine();

}

Console.WriteLine("Hanok Shrestha");

}

}

1. Write a program to ﬁnd the sum of the series 5 +55 + 555 + 5555 + .. n terms. Test Data :

using System; class Program

{

static void Main()

{

Console.Write("Input the number of terms: ");

int numberOfTerms = int.Parse(Console.ReadLine());

Console.Write("Input number: ");

int number = int.Parse(Console.ReadLine());

int sum = 0;

int term = number;

for (int i = 1; i <= numberOfTerms; i++)

{

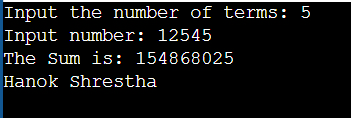
sum += term;

term = (term \* 10) + number;

}

Console.WriteLine("The Sum is: " + sum); Console.WriteLine("Hanok Shrestha");

}

}

1. Write a program to display a pattern like a diamond.

using System; class Program

{

static void Main()

{

int rows;

Console.WriteLine("Enter the number of rows:"); rows = int.Parse(Console.ReadLine());

// Upper half of the diamond for (int i = 1; i <= rows; i++)

{

for (int j = 1; j <= rows - i; j++)

{

Console.Write(" ");

}

for (int k = 1; k <= 2 \* i - 1; k++)

{

Console.Write("\*");

}

Console.WriteLine();

}

// Lower half of the diamond for (int i = rows - 1; i >= 1; i--)

{

for (int j = 1; j <= rows - i; j++)

{

Console.Write(" ");

}

for (int k = 1; k <= 2 \* i - 1; k++)

{

Console.Write("\*");

}

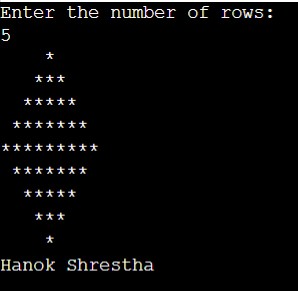
Console.WriteLine();

}

Console.WriteLine("Hanok Shrestha");

}

}



# Tutorial\_4

## : The employee list for a company contains employee code, name, designation and basic pay. The employee is given a house rent allowance (HRA) of 10% of the basic pay and dearness allowance (DA) of 45% of the basic pay. The total pay of the employee is calculated as Basic Pay + HRA + DA. Write a class to define the details of the employee. Write a constructor to assign the required initial values. Add a method to calculate HRA, DA and total pay and print them. Write another class with main method. Create objects for three different employees and calculate HRA, DA and total pay.

**CODE:**

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_4

{

internal class q1

{

class Employee

{

// Properties

public int EmployeeCode { get; } public string Name { get; } public string Designation { get; } public double BasicPay { get; }

public double HRA { get; private set; } public double DA { get; private set; } public double TotalPay { get; private set; }

// Constructor

public Employee(int employeeCode, string name, string designation, double basicPay)

{

EmployeeCode = employeeCode; Name = name;

Designation = designation;

BasicPay = basicPay;

}

// Calculate HRA, DA, and TotalPay public void CalculateSalary()

{

HRA = 0.10 \* BasicPay; DA = 0.45 \* BasicPay;

TotalPay = BasicPay + HRA + DA;

}

// Print Employee Details

public void PrintDetails()

{

Console.WriteLine("Employee Code: " + EmployeeCode); Console.WriteLine("Name: " + Name); Console.WriteLine("Designation: " + Designation); Console.WriteLine("Basic Pay: " + BasicPay); Console.WriteLine("HRA: " + HRA); Console.WriteLine("DA: " + DA); Console.WriteLine("Total Pay: " + TotalPay); Console.WriteLine();

}

}

class Program

{

static void Main(string[] args)

{

// Create objects for three employees

Employee employee1 = new Employee(101, "Hari", "Manager", 80000); Employee employee2 = new Employee(102, "Ramesh", "Developer", 70000); Employee employee3 = new Employee(103, "Sakul", "Analyst", 60000);

// Calculate HRA, DA, and TotalPay for each employee employee1.CalculateSalary(); employee2.CalculateSalary(); employee3.CalculateSalary();

// Print employee details Console.WriteLine("Employee 1 Details:"); employee1.PrintDetails();

Console.WriteLine("Employee 2 Details:"); employee2.PrintDetails();

Console.WriteLine("Employee 3 Details:"); employee3.PrintDetails();

Console.WriteLine("\n hanok shrestha"); Console.ReadLine();

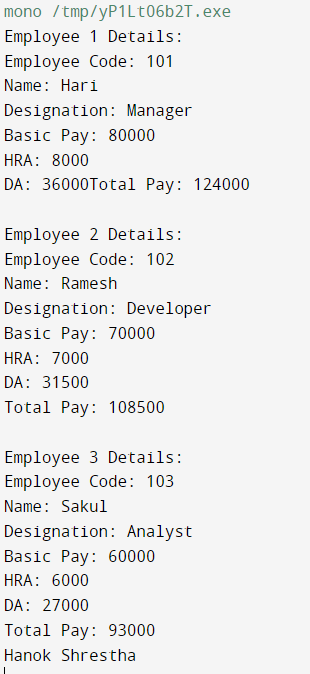
}

}

}

}

## OUTPUT:



1. **: From the following code and given output, complete missing statements and find out error code and correct it.**

## Solution:

//Base class or Parent class. class Shape

{

public double Width; public double Height; public void ShowDim()

{

Console.WriteLine("Width and height are " + Width + " and " + Height);

}

}

// Triangle is derived from Shape.

//Drived class or Child class. class Triangle : Shape

{

public string Style; // style of triangle

// Return area of triangle.

//………………………………Missing statement-1… //

{

return Width \* Height / 2;

}

// Display a triangle's style. public void ShowStyle()

{

//………………………………Missing statement-2… //

}

}

//Driver class which runs the program. class Driver

{

static void Main()

{

Triangle t1 new Triangle(); Triangle t2 new Triangle();

t1.Width =4.0;

t1.Height =4.0; t1.Style ="isosceles"; t2.Width =8.0;

t2.Height =12.0; t2.Style ="right";

Console.WriteLine("Info for t1: "); t1.ShowStyle();

t1.ShowDim();

Console.WriteLine("Area is " + t1.Area());

Console.WriteLine();

Console.WriteLine("Info for t2: "); t2.ShowStyle();

t2.ShowDim();

Console.WriteLine("Area is " + t2.Area());

}

}

Output Info for t1:

Triangle is isosceles

Width and height are 4 and 4 Area is 8

Info for t2:

Triangle is right

Width and height are 8 and 12 Area is 48

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_4

{

internal class q2

{

// Base class or Parent class. class Shape

{

public double Width; public double Height; public void ShowDim()

{

Console.WriteLine("Width and height are " + Width + " and " + Height);

}

}

// Triangle is derived from Shape.

// Drived class or Child class. class Triangle : Shape

{

public string Style; // style of triangle

// Constructor for Triangle public Triangle(string style)

{

Style = style;

}

// Return area of triangle. public double Area()

{

return Width \* Height / 2;

}

// Display a triangle's style. public void ShowStyle()

{

Console.WriteLine("Triangle is " + Style);

}

}

// Driver class which runs the program. class Driver

{

static void Main(string[] args)

{

Triangle t1 = new Triangle("isosceles"); Triangle t2 = new Triangle("right");

t1.Width = 4.0;

t1.Height = 4.0;

t2.Width = 8.0;

t2.Height = 12.0; Console.WriteLine("Info for t1: ");

t1.ShowStyle(); t1.ShowDim();

Console.WriteLine("Area is " + t1.Area());

Console.WriteLine();

Console.WriteLine("Info for t2: "); t2.ShowStyle();

t2.ShowDim();

Console.WriteLine("Area is " + t2.Area());

Console.WriteLine("\n hanok shrestha"); Console.ReadLine();

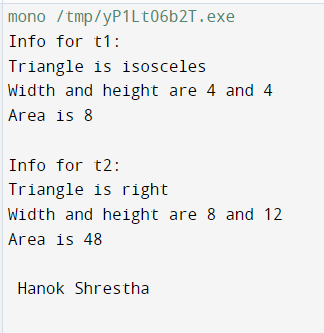
}

}

}

}

## OUTPUT:



**3: Draw a real picture for single level inheritance. Perform following tasks.**

Task 1: Create a class

Task 2: Add few data members as private, protected and public

Task 3: Add few methods as public to work on defined data members

Task 4: Create another applicable class which inherits members from above class Task 5: Add few data members as private, protected and public into second class

Task 6: Add few methods as public to work on defined data members into second class Task 7: Create a Demo class with main method.

Task 8: Create at least two objects of a second class defined in Task 4 into main method and call all methods using that object.

Task 9: Write comment for each important portion of code like data members’ declaration, methods, some important logic etc.

Task 10: Summarize above solution in your own few words to visualize the solution to the end user.

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_4

{

internal class q3

{

// Task 1: Create a class class Shape

{

// Task 2: Add few data members as private, protected and public private double length;

protected double width; public string shapeName;

// Task 3: Add few methods as public to work on defined data members public void SetLength(double l)

{

length = l;

}

public double GetLength()

{

return length;

}

public void SetWidth(double w)

{

width = w;

}

public double GetWidth()

{

return width;

}

public void ShowShapeInfo()

{

Console.WriteLine("Shape Name: " + shapeName); Console.WriteLine("Length: " + length); Console.WriteLine("Width: " + width);

}

}

// Task 4: Create another applicable class which inherits members from above class

class Rectangle : Shape

{

// Task 5: Add few data members as private, protected and public into second class private double area;

protected double perimeter; public string color;

// Task 6: Add few methods as public to work on defined data members into second class public void CalculateArea()

{

area = GetWidth() \* GetLength(); // Use GetWidth() method from base class

}

public void CalculatePerimeter()

{

perimeter = 2 \* (GetWidth() + GetLength()); // Use GetWidth() method from base class

}

public void ShowRectangleInfo()

{

Console.WriteLine("Color: " + color); Console.WriteLine("Area: " + area); Console.WriteLine("Perimeter: " + perimeter);

}

}

// Task 7: Create a Demo class with main method class Demo

{

static void Main()

{

// Task 8: Create at least two objects of a second class defined in Task 4 into main method Rectangle rectangle1 = new Rectangle();

Rectangle rectangle2 = new Rectangle();

// Task 9: Write comment for each important portion of code rectangle1.SetLength(5.0);

rectangle1.SetWidth(3.0); // Use SetWidth() method from base class rectangle1.shapeName = "Rectangle";

// Setting values for data members of rectangle2 using public methods from Shape class rectangle2.SetLength(8.0);

rectangle2.SetWidth(4.0); // Use SetWidth() method from base class rectangle2.shapeName = "Rectangle";

// Setting values for data members of rectangle2 using public members from Rectangle class rectangle2.color = "Blue";

rectangle2.CalculateArea(); rectangle2.CalculatePerimeter();

// Task 8: Call all methods using that object. Console.WriteLine("Info for rectangle1:"); rectangle1.ShowShapeInfo(); Console.WriteLine();

// Displaying the information for rectangle2 Console.WriteLine("Info for rectangle2:"); rectangle2.ShowShapeInfo(); rectangle2.ShowRectangleInfo();

Console.WriteLine("\n hanok shrestha"); Console.ReadLine();

}

}

// Task 10: Summarize above solution in your own few words to visualize the solution to the end user.

// We created a base class called "Shape" with private, protected, and public data members and methods.

// Then, we created a derived class called "Rectangle" which inherits from the "Shape" class.

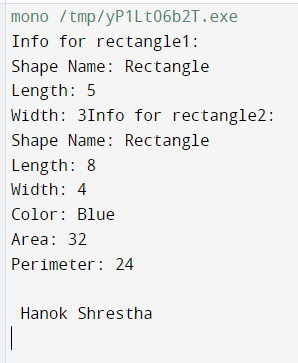
// The "Rectangle" class has its own private, protected, and public data members and methods.

// In the "Demo" class, we create two "Rectangle" objects, set their values, and display the information using various methods.

}

}

## OUTPUT:



1. **: From the following code and given output complete missing statements and find out error code and correct it.**

using System;

namespace StaticVarApplication

{

class StaticVar

{

public static int num;

public counting()

{

num++;

}

//………………………………Missing statement //

{

return num;

}

}

class StaticTester

{

static void Main(string[] args)

{

StaticVar s = new StaticVar(); s.count();

s.count();

s.count();

Console.WriteLine("Variable num: {0}", StaticVar.getNum()); Console.ReadKey();

}

}

}

Output: Variable num: 3 **CODE:**

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_4

{

internal class q4

{

class StaticVar

{

public static int num;

public void counting()

{

num++;

}

public static int getNum()

{

return num;

}

}

class StaticTester

{

static void Main(string[] args)

{

StaticVar s = new StaticVar(); s.counting();

s.counting();

s.counting();

Console.WriteLine("Variable num: {0}", StaticVar.getNum()); Console.ReadKey();

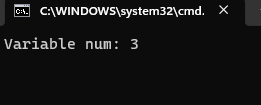
}

}

}

}

## OUTPUT:



1. **: Find out error code and correct it. Print appropriate output as desired.**

using System;

public class A *// This is the base class.*

{

public A(int value)

{

*// Executes some code in the constructor.*

Console.WriteLine("Base constructor A()");

}

}

public class B : A *// This class derives from the previous class.*

{

public B(int value) : base(value)

{

*// The base constructor is called first.*

*// ... Then this code is executed.*

Console.WriteLine("Derived constructor B()");

}

}

class Program

{

static void Main()

{

*// Create a new instance of class A, which is the base class.*

*// ... Then create an instance of B, which executes the base constructor.*

A a = A(0);

B b = B(1);

}

}

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_4

{

internal class q5

{

public class A // This is the base class.

{

public A(int value)

{

// Executes some code in the constructor. Console.WriteLine("Base constructor A()");

}

}

public class B : A // This class derives from the previous class.

{

public B(int value) : base(value)

{

// The base constructor is called first.

// ... Then this code is executed. Console.WriteLine("Derived constructor B()");

}

}

class Program

{

static void Main(string[] args)

{

// Create a new instance of class A, which is the base class.

// ... Then create an instance of B, which executes the base constructor. A a = new A(0);

B b = new B(1);

Console.WriteLine("\n hanok shrestha"); Console.ReadLine();

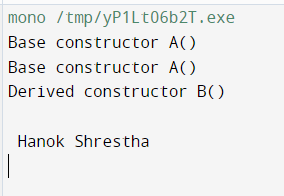
}

}

}

}

## OUTPUT:



1. **: Find out error code and correct it. Print appropriate output as desired. Solution:**

using System; abstract class Test

{

int a;

abstract void A();

}

class Example1 : Test

{

public override void A()

{

Console.WriteLine(*"Example1.A"*); base.a++;

}

}

class Example2 : Test

{

public override void A()

{

Console.WriteLine(*"Example2.A"*); base.a--;

}

}

class Program

{

static void Main()

{

*// Reference Example1 through Test type.*

Test test1 = new Example1(); test1.A();

*// Reference Example2 through Test type.*

Test test2 = new Example2(); test2.A();

}

}

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_4

{

internal class q6

{

abstract class Test

{

protected int a;

public abstract void A();

}

class Example1 : Test

{

public override void A()

{

Console.WriteLine("Example1.A"); a++;

}

}

class Example2 : Test

{

public override void A()

{

Console.WriteLine("Example2.A"); a--;

}

}

class Program

{

static void Main()

{

// Reference Example1 through Test type. Test test1 = new Example1();

test1.A();

// Reference Example2 through Test type. Test test2 = new Example2();

test2.A();

Console.WriteLine("hanok shrestha"); Console.ReadLine();

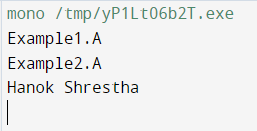
}

}

}

}

## OUTPUT:



1. **: Refer given output and find out error code and correct it.**

sealed class A

{

public int x; public int y;

}

class B : A

{

public int z;

}

class SealedTest2

{

static void Main()

{

A sc = new A(); sc.x = 110;

sc.y = 150;

Console.WriteLine("x = {0}, y = {1}", sc.x, sc.y);

}

}

Output

x = 110, y = 150

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_4

{

internal class q7

{

sealed class A

{

public int x; public int y;

}

class SealedTest2

{

static void Main(string[] args)

{

A sc = new A(); sc.x = 110;

sc.y = 150;

Console.WriteLine("x = {0}, y = {1}", sc.x, sc.y);

Console.WriteLine("\n hanok shrestha"); Console.ReadLine();

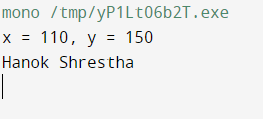
}

}

}

}

## OUTPUT:



1. **: Find out error code and correct it. Print appropriate output as desired. Solution:**

class X

{

protected virtual void F() { Console.WriteLine("X.F"); } protected virtual void F2() { Console.WriteLine("X.F2"); }

}

class Y : X

{

sealed protected override void F() { Console.WriteLine("Y.F"); } protected override void F2() { Console.WriteLine("Y.F2"); }

}

class Z : Y

{

// Overriding F

protected override void F() { Console.WriteLine("Z.F"); }

// Overriding F2

protected override void F2() { Console.WriteLine("Z.F2"); }

}

class SealedMethodTest

{

static void Main()

{

X Obj1 = new X(); Obj1.F();

Obj2.F2();

Y Obj2 = new Y(); Obj2.F();

Obj2.F2();

Z Obj3 = new Z(); Obj3.F();

Obj3.F2();

}

}

## CODE:

using System;

class X

{

protected virtual void F() { Console.WriteLine("X.F"); } protected virtual void F2() { Console.WriteLine("X.F2"); }

}

class Y : X

{

sealed protected override void F() { Console.WriteLine("Y.F"); } protected override void F2() { Console.WriteLine("Y.F2"); }

// New public method to call the protected F2() method public void CallF2()

{

F2();

}

}

class Z : Y

{

// Overriding F

protected override void F() { Console.WriteLine("Z.F"); }

// Overriding F2

protected override void F2() { Console.WriteLine("Z.F2"); }

}

class SealedMethodTest

{

static void Main()

{

X Obj1 = new X(); Obj1.F();

Obj1.F2();

Y Obj2 = new Y(); Obj2.F();

Obj2.CallF2(); // Using the public method to call the protected F2() method

Z Obj3 = new Z(); Obj3.F();

Obj3.F2();

}

}

## OUTPUT:

1. **: This program will throw an exception. Add try, catch and finally blocks to handle this exception.**

using System; class MyClient

{

public static void Main()

{

int x = 0;

int div = 100/x; Console.WriteLine(div);

}

}

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_4

{

internal class q9

{

class MyClient

{

public static void Main()

{

int x = 0; try

{

int div = 100 / x; Console.WriteLine(div);

}

catch (DivideByZeroException ex)

{

Console.WriteLine("Error: " + ex.Message);

}

finally

{

Console.WriteLine("Finally block executed.");

}

Console.WriteLine("\n hanok shrestha"); Console.ReadLine();

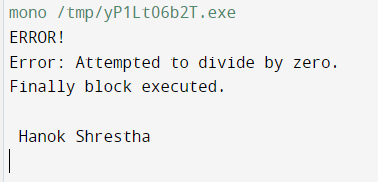
}

}

}

}

## OUTPUT:



**10: Arrange the code to get desirable output**

class MyException : Exception

{

}

class MyClient

{

public static void Main()

{

try

{

}

throw new MyException("my exception generated.");

catch(Exception e)

{

}

Console.WriteLine("LAST STATEMENT");

}

using System;

public MyException(string str)

{

Console.WriteLine("User defined exception"); Console.WriteLine("Exception caught here: " + e.ToString());

}

}

## Output:

Exception caught here: my exception generated. LAST STATEMENT

## CODE:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tutorial\_4

{

internal class q10

{

class MyException : Exception

{

public MyException(string message) : base(message)

{

}

}

class MyClient

{

public static void Main(string[] args)

{

try

{

throw new MyException("my exception generated.");

}

catch (Exception e)

{

Console.WriteLine("Exception caught here: " + e.Message);

}

finally

{

Console.WriteLine("LAST STATEMENT");

}

Console.WriteLine("\nhanok shrestha"); Console.ReadLine();

}

}

}

}

## OUTPUT:

