* Static variables retain their value throughout the program.
* You need to define a static function, to manipulate and use the values of static variables.
* The static keyword ensures that only one instance of a given variable exists for a class.
* Static variables:
  + are used to define constants.
  + can be initialized outside the member function or class definition.
  + have only one copy of the variable existing in the memory for all the objects of that class.
* Static functions:
  + can access only static variables.
  + can be used to check whether an object of a class has been created.
  + exist even before the object is created.

using System;

public class StaticExample

{

public static int s;

public void count()//static variable can be accesed in a non static method

{

s++;

}

public static int display()

{

return s;

}

 }

class Program

{

static int Main(string[] args)

{

StaticExample s = new StaticExample();

s.count();

s.count();

s.count();

Console.WriteLine("The value of variable is {0}",StaticExample.display());

return 0;

}

}

Predict the Output

using System;

static class demo{

int x = 8;//non static variables cannot be accesed from static methods

public static int increment(){

x++;

return (x);}

public static void Main(){

increment();

Console.WriteLine(x);

increment();

Console.WriteLine(x);

Console.ReadLine();

}

}

* + The preceding code will result in an error, because you cannot create instance members in a static class.

What is information Hiding

* Information hiding is a technique of limiting the access to variables, methods, and classes in a program.
* To implement information hiding in a program, you can use the access specifiers.
* The public access specifier:
  + allows a class to share its members with other classes (within or outside the assembly in which the class is defined).
* The private access specifier:
  + allows a class to hide its member variables and member functions from other class objects and functions