# Programming in C# Lab BCA-DS-651

## Manav Rachna International Institute of Research and Studies

### School of Computer Applications

| | |
|---|---|
| | Submitted By |
| Student Name | Garima Chauhan |
| Roll No | 22/FCA/BCA/012 |
| Programme | Bachelor of Computer Applications |
| Semester | 6th Semester |
| Section | A |
| Department | School Of Computer Applications |
| Batch | 2022-25 |
| | |
| | Submitted To |
| Faculty Name | Dr. Neerja Negi |

MANAV RACHNA vidyanatariksha

NAAC A++ ACCREDITED

Manav Rachna Campus Rd, Gadakhor Basti Village, Sector 43, Faridabad, Haryana 121004

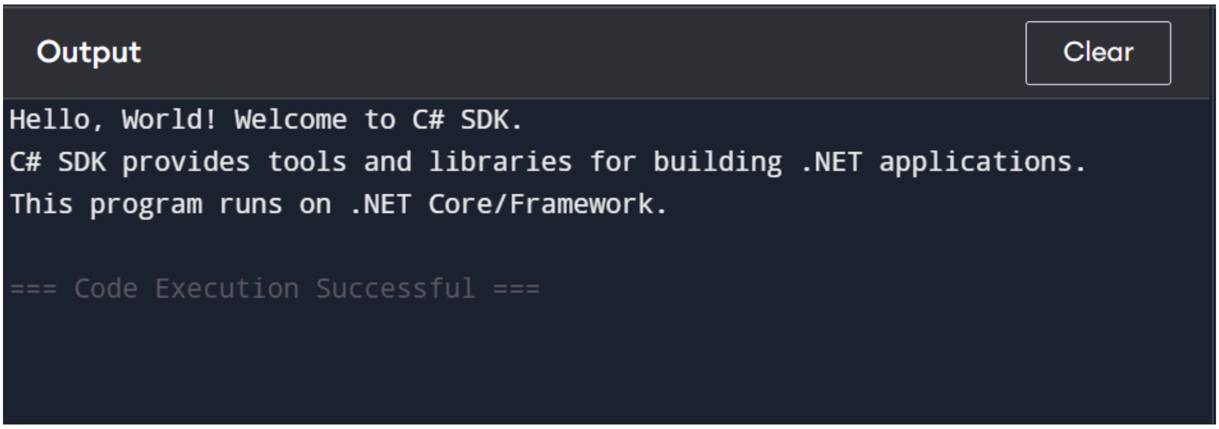
# INDEX

| | | | | |
|---|---|---|---|---|
| 9 | 13/02/2025 | Program 9: Write programs to implement one dimensional and two-dimensional arrays. | | |
| 10 | 14/02/2025 | Program 10: Write programs to understand the working of predefined string functions like Compare (), CompareTo( ), Concat( ), a.Copy( ) and Join(). | | |
| 11 | 27/02/2025 | Program 11: Write a program to implement class and its objects. | | |
| 12 | 27/02/2025 | Program 12: Write a program to implement constructors. | | |

## 1. Write program to demonstrate the working of C# SDK.

```
using System;

class Program

{

static void Main()

{

Console.WriteLine("Hello, World! Welcome to C# SDK.");

Console.WriteLine("C# SDK provides tools and libraries for building .NET applications.");

Console.WriteLine("This program runs on .NET Core/Framework.");

}

}
```

## Output:-

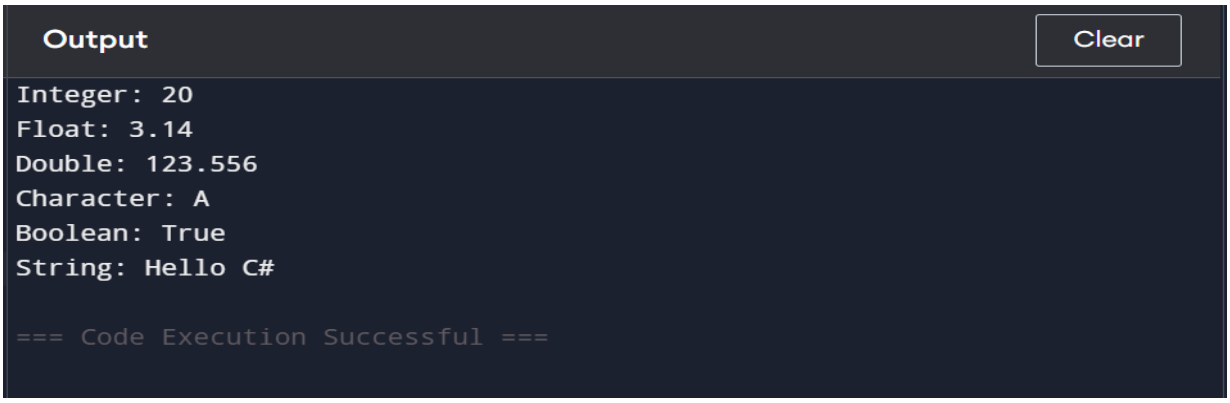


```
Hello, World! Welcome to C# SDK.
C# SDK provides tools and libraries for building .NET applications.
This program runs on .NET Core/Framework.

=== Code Execution Successful ===
```

2. **Write program to show the use of various data types available in C#.**

```
using System;

class DataTypesExample
{
    static void Main()
    {
        int num = 20;
        float pi = 3.14f;
        double largeDecimal = 123.556;
        char letter = 'A';
        bool isTrue = true;
        string message = "Hello C#";

        Console.WriteLine($"Integer: {num}");
        Console.WriteLine($"Float: {pi}");
        Console.WriteLine($"Double: {largeDecimal}");
        Console.WriteLine($"Character: {letter}");
        Console.WriteLine($"Boolean: {isTrue}");
        Console.WriteLine($"String: {message}");
    }
}
```

**Output:-**

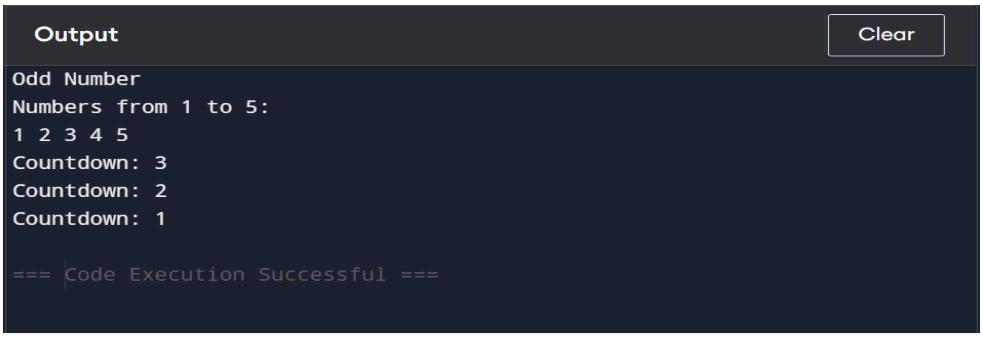


```
Integer: 20
Float: 3.14
Double: 123.556
Character: A
Boolean: True
String: Hello C#

=== Code Execution Successful ===
```

### 3. Write programs to understand the use of Control statements.

```
using System;

class ControlStatementsExample
{
   static void Main()
   {
      int num = 5;

      // If-else
      if (num % 2 == 0)
         Console.WriteLine("Even Number");
      else
         Console.WriteLine("Odd Number");

      // For loop
      Console.WriteLine("Numbers from 1 to 5:");
      for (int i = 1; i <= 5; i++)
         Console.Write(i + " ");
      Console.WriteLine();

      // While loop
      int count = 3;
      while (count > 0)
      {
         Console.WriteLine("Countdown: " + count);
         count--;
      }
   }
}
```

**Output:-**

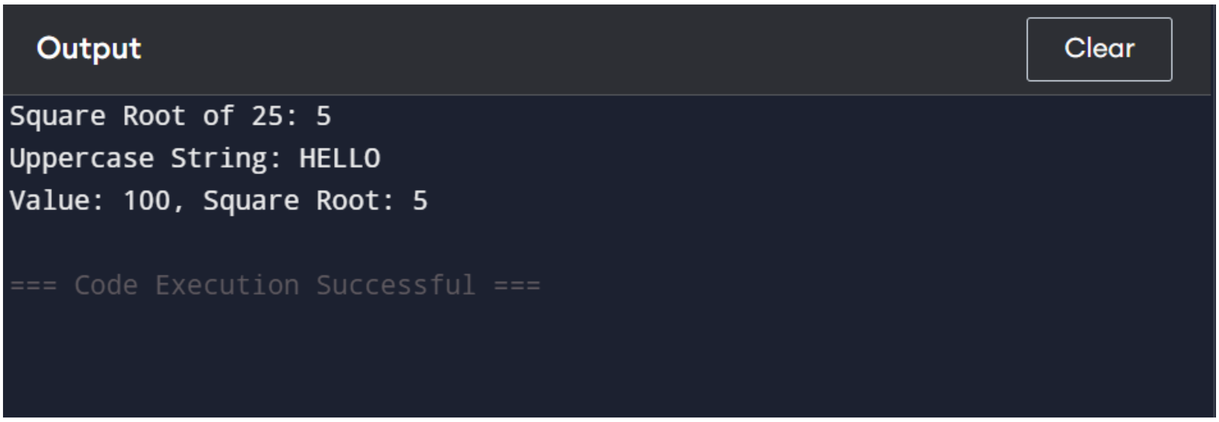


```
Odd Number
Numbers from 1 to 5:
1 2 3 4 5
Countdown: 3
Countdown: 2
Countdown: 1

=== Code Execution Successful ===
```

## 4. Write programs to understand the use of library functions.

```
using System;

class LibraryFunctionsExample
{
    static void Main()
    {
        double sqrtValue = Math.Sqrt(25);
        string upperCase = "hello".ToUpper();
        string formatted = string.Format("Value: {0}, Square Root: {1}", 100, sqrtValue);

        Console.WriteLine($"Square Root of 25: {sqrtValue}");
        Console.WriteLine($"Uppercase String: {upperCase}");
        Console.WriteLine(formatted);
    }
}
```

**Output:-**

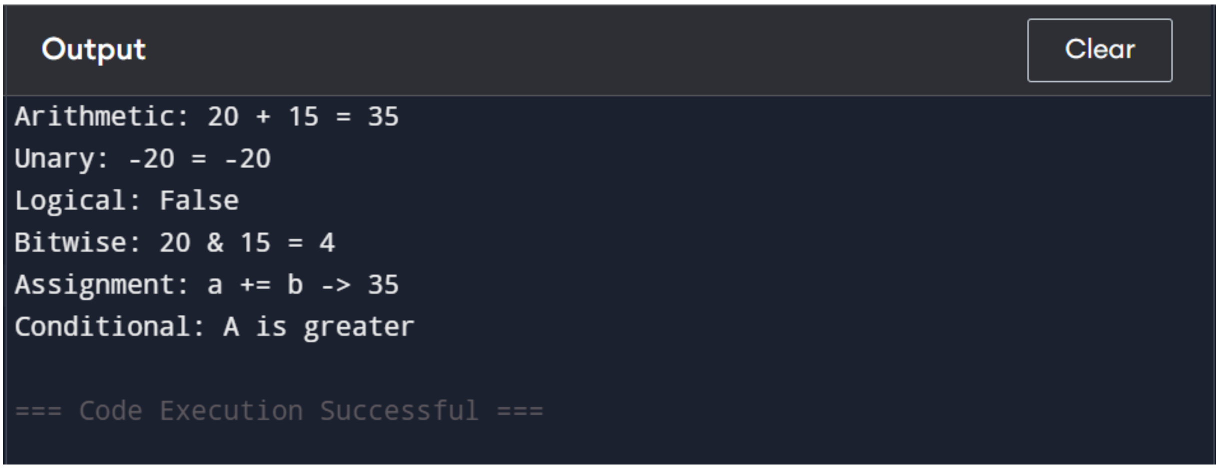


```
Square Root of 25: 5
Uppercase String: HELLO
Value: 100, Square Root: 5

=== Code Execution Successful ===
```

**5. Write a program to demonstrate the use of various arithmetic, unary, logical, bit-wise, assignment and conditional operators.**

```
using System; class
OperatorsExample
{
  static void Main()
  {
    int a = 20, b = 15;
    Console.WriteLine($"Arithmetic: {a} + {b} = {a + b}");
    Console.WriteLine($"Unary: -{a} = {-a}");
    Console.WriteLine($"Logical: {a > b && b < 15}");
    Console.WriteLine($"Bitwise: {a} & {b} = {a & b}");
    Console.WriteLine($"Assignment: a += b -> {a += b}");
    Console.WriteLine($"Conditional: {(a > b ? "A is greater" : "B is greater")}");
  }
}
```

**Output:-**

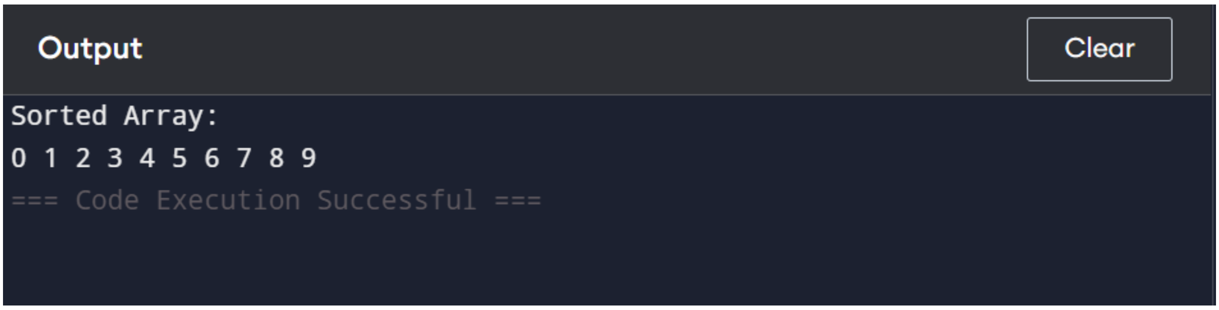


```
Arithmetic: 20 + 15 = 35
Unary: -20 = -20
Logical: False
Bitwise: 20 & 15 = 4
Assignment: a += b -> 35
Conditional: A is greater

=== Code Execution Successful ===
```

**6. Write a program to store 10 elements in an array and display the array elements in increasing order.**

```
using System;

class ArraySorting

{

   static void Main()

   {

      int[] arr = { 9, 5, 2, 8, 3, 7, 1, 6, 4, 0 };

      Array.Sort(arr);

      Console.WriteLine("Sorted Array:");

foreach (int num in arr)

         Console.Write(num + " ");

   }

}
```

**Output:-**

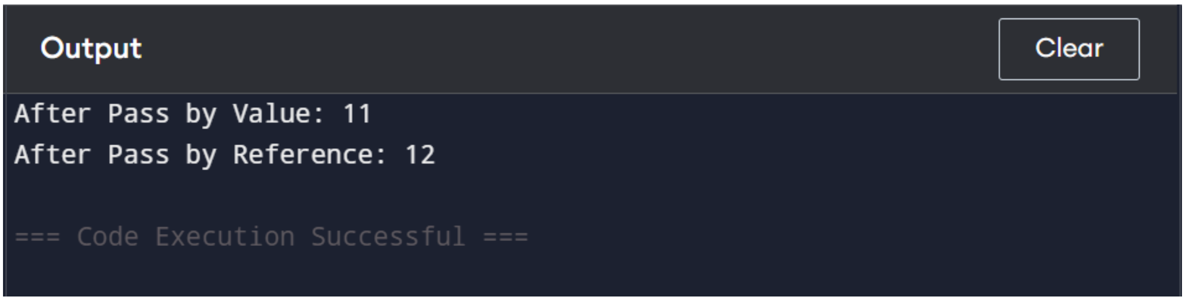


```
Sorted Array:
0 1 2 3 4 5 6 7 8 9
=== Code Execution Successful ===
```

## 7. Demonstrate the use of pass by value and pass by reference by writing a program.

```
 using System;
 class PassExample
{
   static void IncrementByValue(int num)
   {
      num++;
   }
   static void IncrementByReference(ref int num)
   {
      num++;
   }
   static void Main()
   {
      int val = 11;
      IncrementByValue(val);
      Console.WriteLine("After Pass by Value: " + val);
      IncrementByReference(ref val);
      Console.WriteLine("After Pass by Reference: " + val);
   }
}
```

**Output:-**

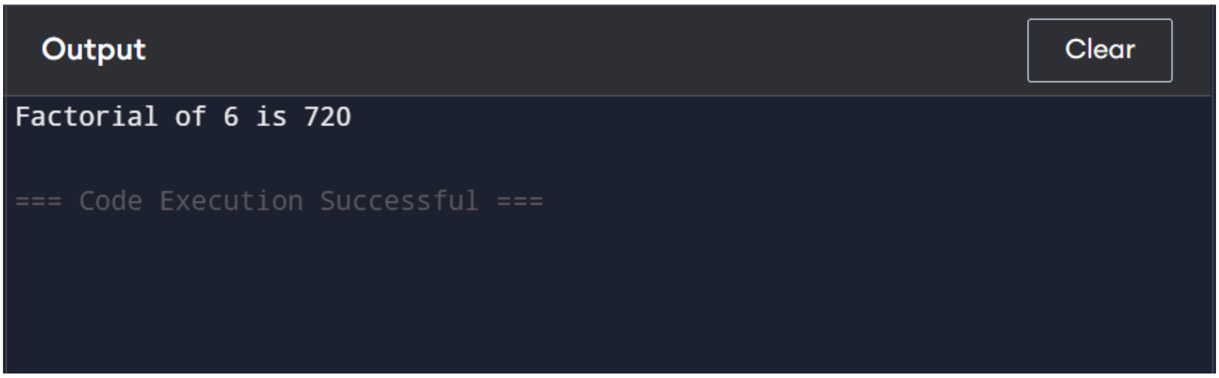


```
After Pass by Value: 11
After Pass by Reference: 12

=== Code Execution Successful ===
```

**8. Write a program to implement recursion.**

```
using System;

class

RecursionExample

{

static int Factorial(int n)

   {

       if (n == 1) return 1;

return n * Factorial(n - 1);

   }

   static void Main()

   {

       int num = 6;

       Console.WriteLine($"Factorial of {num} is {Factorial(num)}");

   }

}
```

**Output:-**



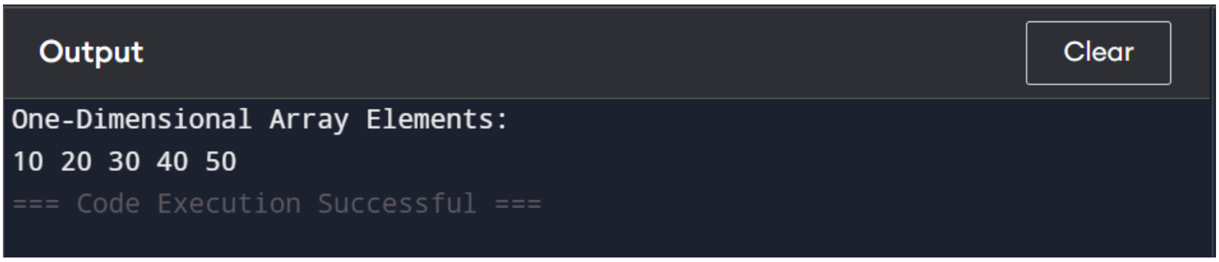
```
Factorial of 6 is 720

=== Code Execution Successful ===
```

## 9. Write programs to implement one dimensional and two-dimensional arrays.

1-D Array

```
using System;

class

OneDArrayExample

{

    static void Main()

    {

        int[] numbers = { 10, 20, 30, 40, 50 };

        Console.WriteLine("One-Dimensional Array Elements:");

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

        {

            Console.Write(numbers[i] + " ");

        }

    }

}
```

### Output:-

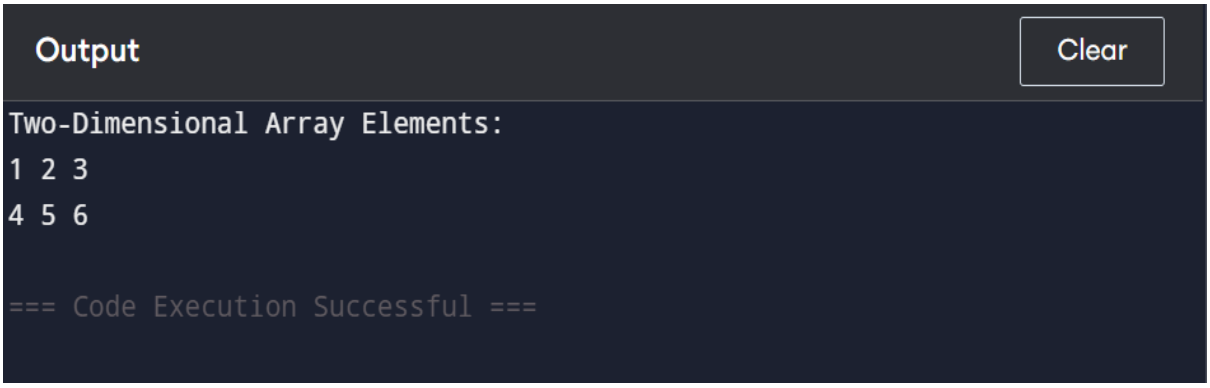


```
One-Dimensional Array Elements:
10 20 30 40 50
=== Code Execution Successful ===
```

# 2-D Array

```
using System;

class

TwoDArrayExample

{

   static void Main()

   {

      int[,] matrix = { { 1, 2, 3 }, { 4, 5, 6 } };

      Console.WriteLine("Two-Dimensional Array Elements:");

      for (int i = 0; i < 2; i++)

      {

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

         {

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

         }

         Console.WriteLine();

      }

   }

}
```

**Output:-**



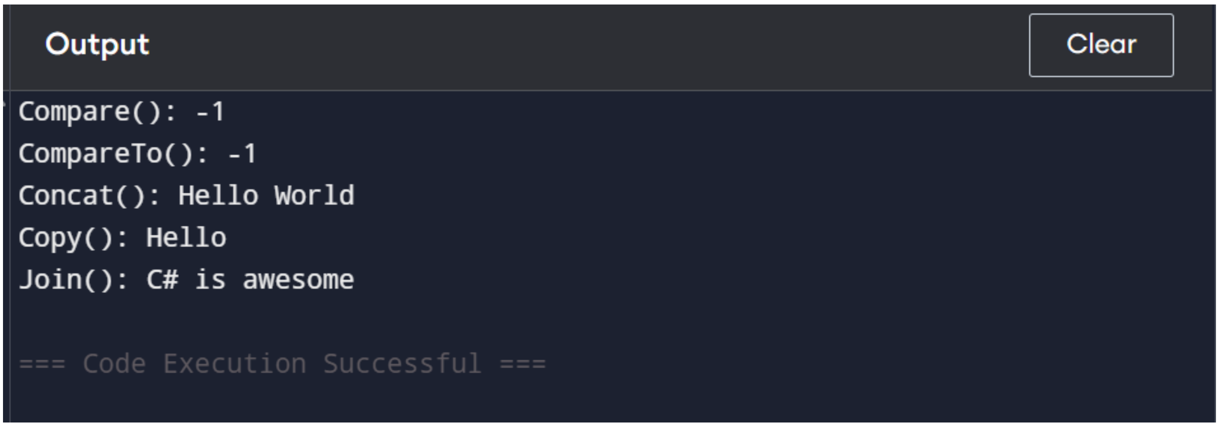
```
Two-Dimensional Array Elements:
1 2 3
4 5 6

=== Code Execution Successful ===
```

## 10. Write programs to understand the working of predefined string functions like Compare( ), CompareTo( ), Concat( ),

a. **Copy( ) and Join().**

```
using System;

class StringFunctions

{

    static void Main()

    {

        string str1 = "Hello";

        string str2 = "World";


        // Compare()

        int result = string.Compare(str1, str2);

        Console.WriteLine("Compare(): " + result); // Returns -1, 0, or 1


        // CompareTo()

        int result2 = str1.CompareTo(str2);

        Console.WriteLine("CompareTo(): " + result2); // Similar to Compare()


        // Concat()

        string concatenated = string.Concat(str1, " ", str2);

        Console.WriteLine("Concat(): " + concatenated);


        // Copy() (Deprecated in newer C# versions)
        string copiedString = string.Copy(str1); // Not recommended in .NET Core and later

        Console.WriteLine("Copy(): " + copiedString);


        // Join()

        string[] words = { "C#", "is", "awesome" };

        string joinedString = string.Join(" ", words);

        Console.WriteLine("Join(): " + joinedString);

    }

}
```

**Output:-**

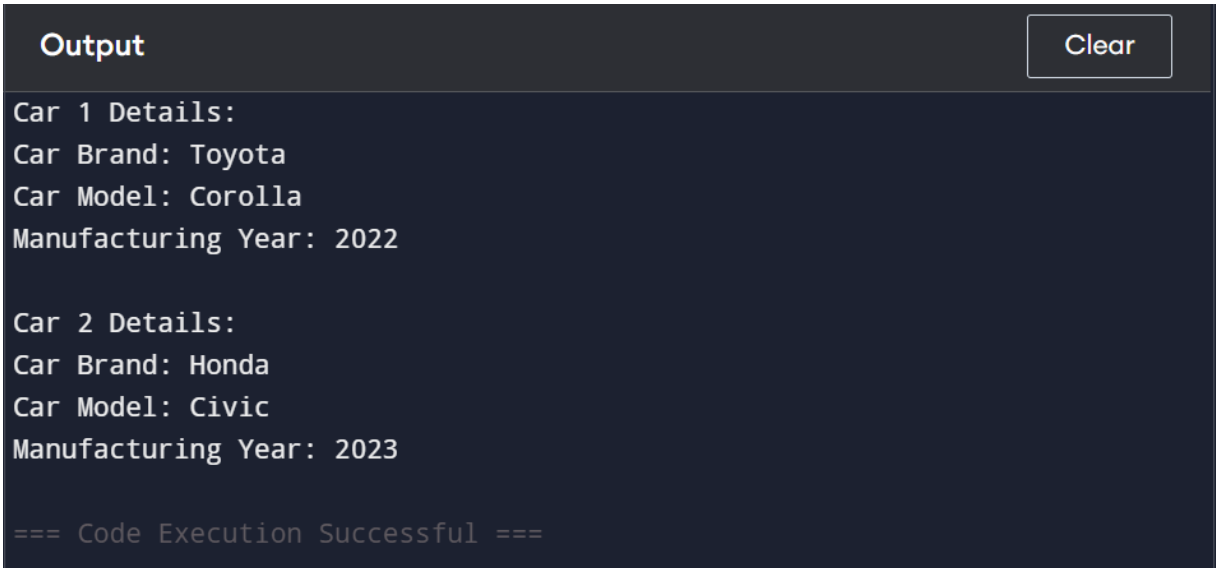


```
Compare(): -1
CompareTo(): -1
Concat(): Hello World
Copy(): Hello
Join(): C# is awesome

=== Code Execution Successful ===
```

## 11. Write a program to implement class and its objects.

```
using System;

class Car
{
    // Fields (Attributes)
    public string Brand;
    public string Model;
    public int Year;

    // Constructor to initialize values
    public Car(string brand, string model, int year)
    {
        Brand = brand;
        Model = model;
        Year = year;
    }

    // Method to display car information
    public void DisplayCarInfo()
    {
        Console.WriteLine("Car Brand: " + Brand);
        Console.WriteLine("Car Model: " + Model);
        Console.WriteLine("Manufacturing Year: " + Year);
    }
}

class Program
{
    static void Main()
    {
        // Creating objects of the Car class
        Car car1 = new Car("Toyota", "Corolla", 2022);
        Car car2 = new Car("Honda", "Civic", 2023);
        // Displaying car details
        Console.WriteLine("Car 1 Details:");
        car1.DisplayCarInfo();

        Console.WriteLine("\nCar 2 Details:");
        car2.DisplayCarInfo();
    }
}
```

**Output:-**

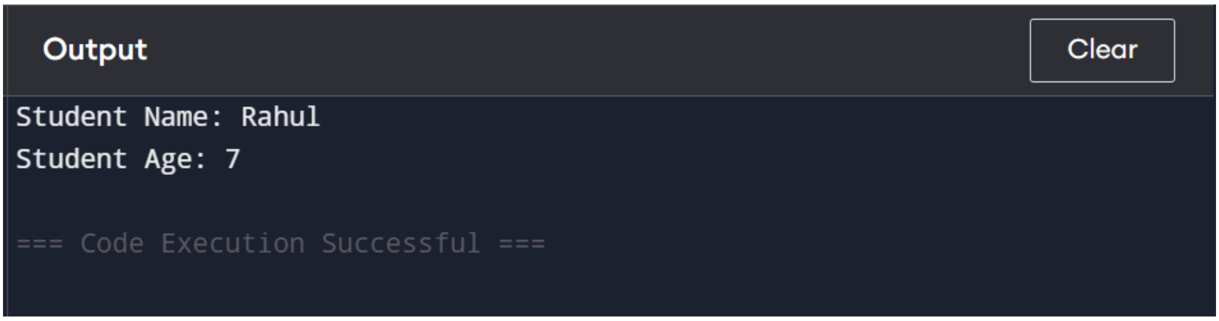


```
Car 1 Details:
Car Brand: Toyota
Car Model: Corolla
Manufacturing Year: 2022

Car 2 Details:
Car Brand: Honda
Car Model: Civic
Manufacturing Year: 2023

=== Code Execution Successful ===
```

## 12. Write a program to implement constructors.

```
using System;

class Student

{

   public string Name;

public int Age;     // Constructor

public Student(string name, int age)

   {

      Name = name;

      Age = age;

   }

   public void Display()

   {

      Console.WriteLine("Student Name: " + Name);

      Console.WriteLine("Student Age: " + Age);

   }

}

class Program

{

   static void Main()

   {

      Student student1 = new Student("Rahul", 07);

student1.Display();

   }

}
```

**Output:-**



```
Student Name: Rahul
Student Age: 7

=== Code Execution Successful ===
```