

Programming in C# LAB

BCA-DS-651

**Manav Rachna International Institute of Research and
Studies**

School of Computer Applications

Submitted By	
Student Name	Dhiraj Kumar
Roll No	22/FCA/BCA/010
Programme	Bachelor of Computer Applications
Semester	6 th Semester
Section	A
Department	School Of Computer Applications
Batch	2022-25
Submitted To	
Faculty Name	Dr. Neerja Negi



**FACULTY OF
COMPUTER APPLICATIONS**

INDEX

S. No.	Date	Aim of the Experiment	Signature	Grade
1	30/01/2025	Program 1: Write program to demonstrate the working of C# SDK.		
2	30/01/2025	Program 2: Write program to show the use of various data types available in C#.		
3	31/01/2025	Program 3: Write programs to understand the use of Control statements.		
4	31/01/2025	Program 4: Write programs to understand the use of library Functions.		
5	05/02/2025	Program 5: Write a program to demonstrate the use of various arithmetic, unary, logical, bit-wise, assignment and conditional operators.		
6	06/02/2025	Program 6: Write a program to store 10 elements in an array and display the array elements in increasing order.		
7	06/02/2025	Program 7: Demonstrate the use of pass by value and pass by reference by writing a program.		
8	13/02/2025	Program 8: Write a program to implement recursion.		
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.		

Programming in C# Lab

Program File

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

Code:

```
using System;
using System.Net.Http;
using System.Threading.Tasks;
class Program
{
    static async Task Main()
    {
        // Create an HTTP Client
        HttpClient client = new HttpClient();
        // Define the URL
        string url = "https://www.example.com";
        // Send a request and get response
        HttpResponseMessage response = await client.GetAsync(url);
        // Read response content
        string content = await response.Content.ReadAsStringAsync();
        // Display part of the content
        Console.WriteLine("==== Website Content (First 100 Characters) =====");
        Console.WriteLine(content.Substring(0, Math.Min(100, content.Length)));
        Console.WriteLine("\nData fetched successfully using System.Net.Http SDK!");
    }
}
```

Output:

```
==== Website Content (First 100 Characters) ====  
<!doctype html>  
<html>  
<head>  
    <title>Example Domain</title>  
  
    <meta charset="utf-8" />  
    <m  
  
Data fetched successfully using System.Net.Http SDK!
```

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

Code:

```
using System;

class Program
{
    static void Main()
    {
        // Integer data types
        int intNumber = 100;

        long longNumber = 123456789L;

        short shortNumber = 32000;

        byte byteNumber = 255;

        // Floating-point data types
        float floatNumber = 10.5f;

        double doubleNumber = 20.99;

        decimal decimalNumber = 100.123456789m;

        // Character and Boolean data types
        char letter = 'A';

        // String data type
        string message = "Hello, C# Programming!";

        // Object data type (can store any type of value)
        object obj1 = 50;

        object obj2 = "C# is powerful!";

        // Displaying all data types
        Console.WriteLine("==== C# Data Types Demonstration =====");

        Console.WriteLine($"Integer (int): {intNumber}");

        Console.WriteLine($"Long (long): {longNumber}");

        Console.WriteLine($"Short (short): {shortNumber}");
```

Code:

```
Console.WriteLine($"Byte (byte): {byteNumber}");  
Console.WriteLine($"Float (float): {floatNumber}");  
Console.WriteLine($"Double (double): {doubleNumber}");  
Console.WriteLine($"Decimal (decimal): {decimalNumber}");  
Console.WriteLine($"Character (char): {letter}");  
Console.WriteLine($"String (string): {message}");  
Console.WriteLine($"Object (int stored in object): {obj1}");  
Console.WriteLine($"Object (string stored in object): {obj2}");  
}  
}
```

Output:

```
===== C# Data Types Demonstration =====  
Integer (int): 100  
Long (long): 123456789  
Short (short): 32000  
Byte (byte): 255  
Float (float): 10.5  
Double (double): 20.99  
Decimal (decimal): 100.123456789  
Character (char): A  
String (string): Hello, C# Programming!  
Object (int stored in object): 50  
Object (string stored in object): C# is powerful!
```

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

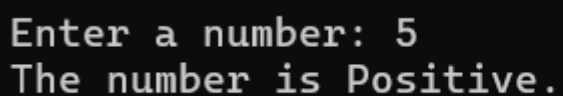
Code (*Conditional Statement (if-else)*):

```
using System;

class Program
{
    static void Main()
    {
        Console.Write("Enter a number: ");
        int number = Convert.ToInt32(Console.ReadLine());

        // Using if-else statement
        if (number > 0)
        {
            Console.WriteLine("The number is Positive.");
        }
        else if (number < 0)
        {
            Console.WriteLine("The number is Negative.");
        }
        else
        {
            Console.WriteLine("The number is Zero.");
        }
    }
}
```

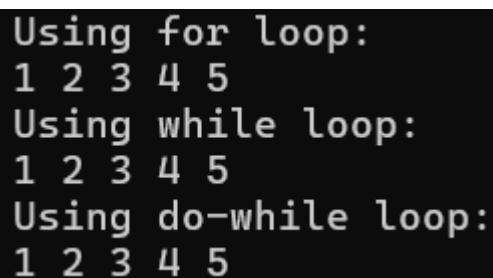
Output (*Conditional Statement (if-else)*):

A screenshot of a terminal window showing the output of the program. The text is displayed in a light green monospace font on a black background. It shows the prompt 'Enter a number: 5' followed by the output 'The number is Positive.' on the next line.

```
Enter a number: 5
The number is Positive.
```


Code (Looping Statements (for, while, do-while)):

```
using System;
class Program
{
    static void Main()
    {
        // Using for loop
        Console.WriteLine("Using for loop:");
        for (int i = 1; i <= 5; i++)
        {
            Console.Write(i + " ");
        }
        Console.WriteLine("\nUsing while loop:");
        int j = 1;
        while (j <= 5)
        {
            Console.Write(j + " ");
            j++;
        }
        Console.WriteLine("\nUsing do-while loop:");
        int k = 1;
        do
        {
            Console.Write(k + " ");
            k++;
        } while (k <= 5);
    }
}
```

Output (Conditional Statement (if-else)):

```
Using for loop:
1 2 3 4 5
Using while loop:
1 2 3 4 5
Using do-while loop:
1 2 3 4 5
```

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

Code:

```
using System;

class Program
{
    static void Main()
    {
        double number = -25.75;

        // Absolute value
        Console.WriteLine("Absolute value: " + Math.Abs(number));

        // Square root
        Console.WriteLine("Square root of 25: " + Math.Sqrt(25));

        // Power function
        Console.WriteLine("2 raised to the power 3: " + Math.Pow(2, 3));

        // Rounding functions
        Console.WriteLine("Round(25.75): " + Math.Round(number));
        Console.WriteLine("Ceiling(25.75): " + Math.Ceiling(number));
        Console.WriteLine("Floor(25.75): " + Math.Floor(number));
    }
}
```

Output:

```
Absolute value: 25.75
Square root of 25: 5
2 raised to the power 3: 8
Round(25.75): -26
Ceiling(25.75): -25
Floor(25.75): -26
```

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

Code:

```
using System;

class Program
{
    static void Main()
    {
        int a = 10, b = 5, result;

        // Arithmetic Operators (+ and -)
        result = a + b;
        Console.WriteLine("Addition (a + b) = " + result);
        result = a - b;
        Console.WriteLine("Subtraction (a - b) = " + result);

        // Unary Operators (+ and -)
        Console.WriteLine("Unary Plus (+a) = " + (+a));
        Console.WriteLine("Unary Minus (-a) = " + (-a));

        // Logical Operator (&& and || using simple condition)
        bool x = true, y = false;
        Console.WriteLine("Logical AND (x && y) = " + (x && y));
        Console.WriteLine("Logical OR (x || y) = " + (x || y));

        // Bitwise Operators (& and |)
        Console.WriteLine("Bitwise AND (a & b) = " + (a & b));
        Console.WriteLine("Bitwise OR (a | b) = " + (a | b));

        // Assignment Operators (+= and -=)
        a += b; // Equivalent to a = a + b
        Console.WriteLine("After a += b, a = " + a);
        a -= b; // Equivalent to a = a - b
        Console.WriteLine("After a -= b, a = " + a);
```

Code:

```
// Conditional (Ternary) Operator

string resultText = (a > b) ? "a is greater than b" : "b is greater or equal to a";

Console.WriteLine("Conditional Operator Result: " + resultText);

}

}
```

Output:

```
Addition (a + b) = 15
Subtraction (a - b) = 5
Unary Plus (+a) = 10
Unary Minus (-a) = -10
Logical AND (x && y) = False
Logical OR (x || y) = True
Bitwise AND (a & b) = 0
Bitwise OR (a | b) = 15
After a += b, a = 15
After a -= b, a = 10
Conditional Operator Result: a is greater than b
```

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

Code:

```
using System;
class Program
{
    static void Main()
    {
        int[] numbers = new int[10];
        // Taking input from the user
        Console.WriteLine("Enter 10 elements for the array:");
        for (int i = 0; i < 10; i++)
        {
            Console.Write("Element {0}: ", i + 1);
            numbers[i] = Convert.ToInt32(Console.ReadLine());
        }
        // Sorting the array in increasing order
        Array.Sort(numbers);
        // Displaying sorted elements
        Console.WriteLine("\nArray elements in increasing order:");
        foreach (int num in numbers)
        {
            Console.Write(num + " ");
        }
    }
}
```

Output:

```
Enter 10 elements for the array:
```

```
Element 1: 2
```

```
Element 2: 3
```

```
Element 3: 4
```

```
Element 4: 1
```

```
Element 5: 56
```

```
Element 6: 22
```

```
Element 7: 43
```

```
Element 8: 78
```

```
Element 9: 43
```

```
Element 10: 12
```

```
Array elements in increasing order:
```

```
1 2 3 4 12 22 43 43 56 78
```

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

Code:

```
using System;
class Program
{
    // Pass by Value: The original value remains unchanged
    static void PassByValue(int num)
    {
        num = num + 10; // Modifying the value
        Console.WriteLine("Inside PassByValue method: num = " + num);
    }
    // Pass by Reference: The original value is modified
    static void PassByReference(ref int num)
    {
        num = num + 10; // Modifying the value
        Console.WriteLine("Inside PassByReference method: num = " + num);
    }
    static void Main()
    {
        int a = 5, b = 5;
        Console.WriteLine("Before calling PassByValue: a = " + a);
        PassByValue(a);
        Console.WriteLine("After calling PassByValue: a = " + a);
        Console.WriteLine("\nBefore calling PassByReference: b = " + b);
        PassByReference(ref b);
        Console.WriteLine("After calling PassByReference: b = " + b);
    }
}
```

Output:

```
Before calling PassByValue: a = 5  
Inside PassByValue method: num = 15  
After calling PassByValue: a = 5  
  
Before calling PassByReference: b = 5  
Inside PassByReference method: num = 15  
After calling PassByReference: b = 15
```


8. Write a program to implement recursion.

Code:

```
using System;

class Program
{
    // Recursive function to calculate factorial
    static int Factorial(int n)
    {
        if (n == 0 || n == 1)
            return 1; // Base case
        else
            return n * Factorial(n - 1); // Recursive case
    }

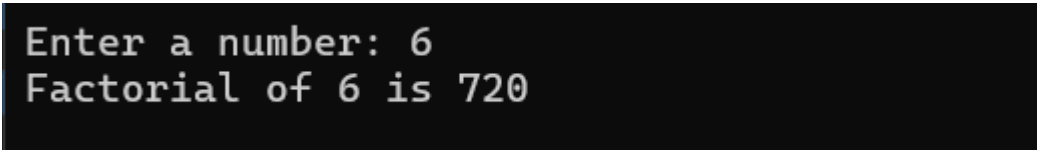
    static void Main()
    {
        Console.Write("Enter a number: ");

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

        int result = Factorial(num); // Function call

        Console.WriteLine("Factorial of {0} is {1}", num, result);
    }
}
```

Output:

A screenshot of a terminal window with a black background and white text. It shows the output of the program: "Enter a number: 6" followed by "Factorial of 6 is 720".

```
Enter a number: 6
Factorial of 6 is 720
```

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

Code:

```
using System;

class ArrayExample
{
    static void Main()
    {
        // *** ONE-DIMENSIONAL ARRAY (1D) ***

        int[] numbers = new int[5]; // Declaring a 1D array

        Console.WriteLine("Enter 5 elements for the 1D array:");
        for (int i = 0; i < 5; i++)
        {
            Console.Write("Element {0}: ", i + 1);
            numbers[i] = Convert.ToInt32(Console.ReadLine());
        }

        // Displaying 1D array elements
        Console.WriteLine("\n1D Array elements:");
        foreach (int num in numbers)
        {
            Console.Write(num + " ");
        }

        Console.WriteLine("\n\n-----\n");

        // *** TWO-DIMENSIONAL ARRAY (2D) ***

        int[,] matrix = new int[2, 3]; // Declaring a 2D array

        Console.WriteLine("Enter 6 elements for the 2x3 matrix:");
        for (int i = 0; i < 2; i++)
        {
            for (int j = 0; j < 3; j++)
```

Code:

```
{
    Console.Write("Element at ({0},{1}): ", i, j);
    matrix[i, j] = Convert.ToInt32(Console.ReadLine());
}
}

// Displaying 2D matrix elements
Console.WriteLine("\n2D Matrix elements:");
for (int i = 0; i < 2; i++)
{
    for (int j = 0; j < 3; j++)
    {
        Console.Write(matrix[i, j] + "\t");
    }

    Console.WriteLine(); // New line after each row
}
}}
```

Output:

```
Enter 5 elements for the 1D array:
Element 1: 4
Element 2: 3
Element 3: 5
Element 4: 6
Element 5: 1

1D Array elements:
4 3 5 6 1

-----

Enter 6 elements for the 2x3 matrix:
Element at (0,0): 4
Element at (0,1): 2
Element at (0,2): 7
Element at (1,0): 3
Element at (1,1): 1
Element at (1,2): 6

2D Matrix elements:
4      2      7
3      1      6
```

10. Write programs to understand the working of predefined string functions like Compare (), CompareTo(), Concat(), a.Copy() and Join().

Code:

```
using System;

class StringFunctionsDemo
{
    static void Main()
    {
        string str1 = "Hello";
        string str2 = "World";
        string str3 = "Hello";

        // Using Compare() - Compares two strings (case-sensitive)
        Console.WriteLine("Compare(str1, str2): " + string.Compare(str1, str2)); // Returns -1, 0, or 1
        Console.WriteLine("Compare(str1, str3): " + string.Compare(str1, str3)); // Returns 0 as both are
        equal

        // Using CompareTo() - Works like Compare() but used with an instance
        Console.WriteLine("str1.CompareTo(str2): " + str1.CompareTo(str2));
        Console.WriteLine("str1.CompareTo(str3): " + str1.CompareTo(str3));

        // Using Concat() - Joins two or more strings
        string concatenated = string.Concat(str1, " ", str2);
        Console.WriteLine("Concat(str1, str2): " + concatenated);

        // Using Copy() - Creates a copy of the string
        string copyOfStr1 = string.Copy(str1);
        Console.WriteLine("Copy of str1: " + copyOfStr1);

        // Using Join() - Joins multiple strings with a separator
        string[] words = { "C#", "is", "fun" };
        string joinedString = string.Join(" ", words);
        Console.WriteLine("Join(words): " + joinedString);
    }
}
```

Output:

```
Compare(str1, str2): -1  
Compare(str1, str3): 0  
str1.CompareTo(str2): -1  
str1.CompareTo(str3): 0  
Concat(str1, str2): Hello World  
Copy of str1: Hello  
Join(words): C# is fun
```

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

Code:

```
using System;

class Student
{
    // Data Members (Variables)
    public string name;
    public int age;

    // Method to display student details
    public void Display()
    {
        Console.WriteLine("Student Name: " + name);
        Console.WriteLine("Student Age: " + age);
    }
}

class Program
{
    static void Main()
    {
        // Creating an object of the Student class
        Student student1 = new Student();
        // Assigning values to object properties
        student1.name = "John";
        student1.age = 20;
        // Calling the method
        Console.WriteLine("Student Details:");
        student1.Display();
    }
}
```

Output:

```
Student Details:  
Student Name: John  
Student Age: 20
```

12. Write a program to implement constructors.

Code:

```
using System;

class Student
{
    // Data Members
    public string name;
    public int age;

    // Constructor (Same name as class, no return type)
    public Student(string studentName, int studentAge)
    {
        name = studentName;
        age = studentAge;
    }

    // Method to display student details
    public void Display()
    {
        Console.WriteLine("Student Name: " + name);
        Console.WriteLine("Student Age: " + age);
    }
}

class Program
{
    static void Main()
    {
        // Creating an object and passing values to the constructor
        Student student1 = new Student("John", 20);

        // Display student details
        Console.WriteLine("Student Details:");
    }
}
```


Code:

```
    student1.Display();  
}  
}
```

Output:

```
Student Details:  
Student Name: John  
Student Age: 20
```