**Console Input:**

1. **Reading from Console:**

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

string input = Console.ReadLine();

Console.WriteLine("You entered: " + input);

Here, **Console.ReadLine()** is used to read a line of text from the console.

### Console Output:

1. **Writing to Console:**

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

This line will print "Hello, World!" to the console.

### Formatting Output:

1. **String Interpolation:**

int age = 25;

Console.WriteLine($"My age is {age}");

String interpolation allows you to embed expressions inside string literals.

1. **Composite Formatting:**

string name = "John";

int age = 30;

Console.WriteLine("Name: {0}, Age: {1}", name, age);

Using placeholders **{0}**, **{1}**, etc., to format and insert values into a string.

### File Input/Output:

1. **Reading from a File:**

string path = "sample.txt";

string content = File.ReadAllText(path);

Console.WriteLine("File Content: " + content);

The **File.ReadAllText** method reads the entire content of a file into a string.

**2.Writing to a File:**

string path = "output.txt";

string content = "This is some content.";

File.WriteAllText(path, content);

The **File.WriteAllText** method writes the specified string to a file.

### Reading Command-Line Arguments:

1. **Command-Line Arguments:**

static void Main(string[] args)

{

foreach (string arg in args)

{

Console.WriteLine("Argument: " + arg);

}

}

Command-line arguments can be accessed through the **args** parameter of the **Main** method.

### Reading Input as Numbers:

1. **Parsing Input as Integer:**

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

string input = Console.ReadLine();

if (int.TryParse(input, out int number))

{

Console.WriteLine("You entered a valid number: " + number);

}

else

{

Console.WriteLine("Invalid input. Not a number.");

}

Using **int.TryParse** to convert user input to an integer.