



Program 1

```
// This would be part of the
WeatherService.SDK namespace
public class WeatherServiceClient
{
    private readonly string _apiKey;

    public WeatherServiceClient(string
apiKey)
    {
        _apiKey = apiKey;
    }

    public async Task<WeatherInfo>
GetWeatherAsync(string city)
    {
        // Simulate an API call
        await Task.Delay(1000); // Simulate
network delay

        // Return mock weather data
        return new WeatherInfo
        {
```





```
}  
}
```

```
public class WeatherInfo  
{  
    public int Temperature { get; set; }  
    public string Condition { get; set; }  
    public int Humidity { get; set; }  
}
```

Program 2

```
using System;
```

```
namespace DataTypesDemo  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            // Integer Types  
            int intValue = 42;  
            long longValue = 12345678901234;  
            short shortValue = 32000;
```





Program 2

using System;

namespace DataTypesDemo

{

class Program

{

static void Main(string[] args)

{

// Integer Types

int intValue = 42;

long longValue = 12345678901234;

short shortValue = 32000;

byte byteValue = 255;

// Floating Point Types

float floatValue = 3.14f;

double doubleValue =

3.14159265358979;

decimal decimalValue = 19.99m;

// Character Type

char charValue = 'A';



```
char charValue = 'A';
```

```
// Boolean Type
```

```
bool boolValue = true;
```

```
// String Type
```

```
string stringValue = "Hello, C#!";
```

```
// Object Type
```

```
object objectValue = "This is an  
object type";
```

```
// Displaying the values
```

```
Console.WriteLine("Integer  
Types:");
```

```
Console.WriteLine($"int:  
{intValue}");
```

```
Console.WriteLine($"long:  
{longValue}");
```

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

```
Console.WriteLine($"byte:  
{byteValue}");
```

```
Console.WriteLine();
```

```
Console.WriteLine("Floating Point
```



```
Console.WriteLine();
```

```
    Console.WriteLine("Floating Point  
Types:");
```

```
    Console.WriteLine($"float:  
{floatValue}");
```

```
    Console.WriteLine($"double:  
{doubleValue}");
```

```
    Console.WriteLine($"decimal:  
{decimalValue}");
```

```
    Console.WriteLine();
```

```
    Console.WriteLine("Character  
Type:");
```

```
    Console.WriteLine($"char:  
{charValue}");
```

```
    Console.WriteLine();
```

```
    Console.WriteLine("Boolean  
Type:");
```

```
    Console.WriteLine($"bool:  
{boolValue}");
```

```
    Console.WriteLine();
```

```
    Console.WriteLine("String Type:");
```

```
    Console.WriteLine($"string:
```



```
//  
    Console.WriteLine($"bool:  
{boolValue}");  
    Console.WriteLine();  
  
    Console.WriteLine("String Type:");  
    Console.WriteLine($"string:  
{stringValue}");  
    Console.WriteLine();  
  
    Console.WriteLine("Object Type:");  
    Console.WriteLine($"object:  
{objectValue}");  
    Console.WriteLine();  
  
    // Demonstrating type conversion  
    Console.WriteLine("Type  
Conversion:");  
    int convertedInt =  
(int)decimalValue; // Implicit conversion  
    Console.WriteLine($"Converted  
decimal to int: {convertedInt}");  
}  
}  
}
```





Program 3

using System;

namespace ControlStatementsDemo

{

class Program

{

static void Main(string[] args)

{

// Conditional Statements

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

int number =

Convert.ToInt32(Console.ReadLine());

if (number > 0)

{

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

}

else if (number < 0)

{

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





```
{  
    Console.WriteLine("The number is  
zero.");  
}
```

// Switch Statement

```
Console.WriteLine("Enter a day  
number (1-7):");  
int day =  
Convert.ToInt32(Console.ReadLine());  
switch (day)  
{  
    case 1:  
        Console.WriteLine("Monday");  
        break;  
    case 2:  
        Console.WriteLine("Tuesday");  
        break;  
    case 3:  
        Console.WriteLine("Wednesday");  
        break;  
    case 4:  
        Console.WriteLine("Thursday");  
        break;  
    case 5:
```





```
Console.WriteLine("Wednesday");
    break;
case 4:
    Console.WriteLine("Thursday");
    break;
case 5:
    Console.WriteLine("Friday");
    break;
case 6:
    Console.WriteLine("Saturday");
    break;
case 7:
    Console.WriteLine("Sunday");
    break;
default:
    Console.WriteLine("Invalid day
number.");
    break;
}
```

// Looping Statements

```
Console.WriteLine("Counting from 1
to 5 using a for loop:");
for (int i = 1; i <= 5; i++)
{
    Console.WriteLine(i);
}
```



```
Console.WriteLine("Counting from 1  
to 5 using a for loop:");
```

```
    for (int i = 1; i <= 5; i++)  
    {  
        Console.WriteLine(i);  
    }
```

```
Console.WriteLine("Counting from 1  
to 5 using a while loop:");
```

```
    int j = 1;  
    while (j <= 5)  
    {  
        Console.WriteLine(j);  
        j++;  
    }
```

```
Console.WriteLine("Counting from 1  
to 5 using a do while loop:");
```

```
    int k = 1;  
    do  
    {  
        Console.WriteLine(k);  
        k++;  
    } while (k <= 5);
```

```
// Foreach Loop
```

```
    JTT,  
}
```

Console.WriteLine("Counting from 1
to 5 using a do while loop:");

```
int k = 1;  
do  
{  
    Console.WriteLine(k);  
    k++;  
} while (k <= 5);
```

// Foreach Loop

Console.WriteLine("Iterating over an
array using foreach:");

```
string[] fruits = { "Apple", "Banana",  
"Cherry" };  
foreach (string fruit in fruits)  
{  
    Console.WriteLine(fruit);  
}
```

// Jump Statements

Console.WriteLine("Using break and
continue in a loop:");

```
for (int m = 1; m <= 10; m++)  
-
```



```
continue in a loop:");
    for (int m = 1; m <= 10; m++)
    {
        if (m == 5)
        {
            Console.WriteLine("Breaking
the loop at 5.");
            break; // Exit the loop when m is
5
        }
        if (m % 2 == 0)
        {
            Console.WriteLine($"Skipping
even number: {m}");
            continue; // Skip the rest of the
loop for even numbers
        }
        Console.WriteLine(m);
    }
}
}
```

Program 4

using System;



Program 4

```
using System;
using System.Collections.Generic;

namespace LibraryFunctionsDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            // String Manipulation Functions
            string originalString = "Hello,
World!";
            Console.WriteLine("Original String:
" + originalString);
            Console.WriteLine("Length of
String: " + originalString.Length);
            Console.WriteLine("Uppercase: " +
originalString.ToUpper());
            Console.WriteLine("Lowercase: " +
originalString.ToLower());
            Console.WriteLine("Substring (7, 5):
" + originalString.Substring(7, 5));
            Console.WriteLine("Contains
'World': " +
```

```
originalString.Contains("Math"),
```

```
    Console.WriteLine();
```

```
    // Math Functions
```

```
    Console.WriteLine("Math  
Functions:");
```

```
    Console.WriteLine("Square Root of  
16: " + Math.Sqrt(16));
```

```
    Console.WriteLine("Power of 2^3: "  
+ Math.Pow(2, 3));
```

```
    Console.WriteLine("Random  
Number: " + new Random().Next(1,  
100)); // Random number between 1 and  
99
```

```
    Console.WriteLine();
```

```
    // Date and Time Functions
```

```
    DateTime now = DateTime.Now;  
    Console.WriteLine("Current Date  
and Time: " + now);
```

```
    Console.WriteLine("Today: " +  
now.ToShortDateString());
```

```
    Console.WriteLine("Current Time: "  
+ now.ToShortTimeString());
```

```
    Console.WriteLine("Add 5 days: " +  
now.AddDays(5));
```

```
    Console.WriteLine();
```



```
List<string> fruits = new List<string> { "Apple", "Banana",  
"Cherry" };
```

```
    Console.WriteLine("Fruits List:");
```

```
    fruits.Add("Date");
```

```
    fruits.Remove("Banana");
```

```
    foreach (var fruit in fruits)
```

```
    {
```

```
        Console.WriteLine(fruit);
```

```
    }
```

```
    Console.WriteLine("Total Fruits: " +  
fruits.Count);
```

```
}
```

```
}
```

```
}
```

Program 5

```
using System;
```

```
namespace OperatorsDemo
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            // ...
```



Program 5

```
using System;
```

```
namespace OperatorsDemo
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            // Arithmetic Operators
```

```
            int a = 10;
```

```
            int b = 5;
```

```
            Console.WriteLine("Arithmetic  
Operators:");
```

```
            Console.WriteLine($"a + b = {a +  
b}"); // Addition
```

```
            Console.WriteLine($"a - b = {a -  
b}"); // Subtraction
```

```
            Console.WriteLine($"a * b = {a *  
b}"); // Multiplication
```

```
            Console.WriteLine($"a / b = {a /  
b}"); // Division
```

```
            Console.WriteLine($"a % b = {a %  
b}"); // Modulus
```

```
Console.WriteLine($"a % b = {a %  
b}"); // Modulus
```

```
Console.WriteLine();
```

```
// Unary Operators
```

```
int c = 5;
```

```
Console.WriteLine("Unary  
Operators:");
```

```
Console.WriteLine($"Initial value of  
c: {c}");
```

```
Console.WriteLine($"c++ = {c++}  
(Post-increment, c is now {c})");
```

```
Console.WriteLine($"++c = {++c}  
(Pre-increment, c is now {c})");
```

```
Console.WriteLine($"c-- = {c--}  
(Post-decrement, c is now {c})");
```

```
Console.WriteLine($"--c = {--c}  
(Pre-decrement, c is now {c})");
```

```
Console.WriteLine($"Unary plus  
(+c) = {+c}"); // Unary plus
```

```
Console.WriteLine($"Unary minus (-  
c) = {-c}"); // Unary minus
```

```
Console.WriteLine();
```

```
// Conditional (Ternary) Operator
```

```
int d = 20;
```