

Selected files

1 printable files

01_DATA_TYPES\02_OPERTORS\Program.cs

01_DATA_TYPES\02_OPERTORS\Program.cs

```
1  using System;
2
3  namespace _02_OPERATORS
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              #region 01. ARITHMETIC OPERATORS
10             Console.WriteLine("01. Arithmetic Operators:");
11
12             int result;
13             // Addition
14             result = 20 + 10;
15             Console.WriteLine($"Addition: 20 + 10 = {result}"); // Output: 30
16
17             // Subtraction
18             result = 20 - 10;
19             Console.WriteLine($"Subtraction: 20 - 10 = {result}"); // Output: 10
20
21             // Multiplication
22             result = 20 * 10;
23             Console.WriteLine($"Multiplication: 20 * 10 = {result}"); // Output: 200
24
25             // Division
26             result = 20 / 10;
27             Console.WriteLine($"Division: 20 / 10 = {result}"); // Output: 2
28
29             // Modulo
30             result = 20 % 10;
31             Console.WriteLine($"Modulo: 20 % 10 = {result}"); // Output: 0
32
33             Console.WriteLine();
34             #endregion
35
36             #region 02. ASSIGNMENT OPERATORS
37             Console.WriteLine("02. Assignment Operators:");
38
39             int x = 15;
40             x += 10;
41             Console.WriteLine($"x += 10 → x = {x}"); // Output: 25
42
43             x = 20;
44             x -= 5;
45             Console.WriteLine($"x -= 5 → x = {x}"); // Output: 15
46
```

```
47     x = 15;
48     x *= 5;
49     Console.WriteLine($"x *= 5 → x = {x}"); // Output: 75
50
51     x = 25;
52     x /= 5;
53     Console.WriteLine($"x /= 5 → x = {x}"); // Output: 5
54
55     x = 25;
56     x %= 5;
57     Console.WriteLine($"x %= 5 → x = {x}"); // Output: 0
58
59     Console.WriteLine();
60     #endregion
61
62     #region 03. RELATIONAL OPERATORS
63     Console.WriteLine("03. Relational Operators:");
64
65     bool resultBool;
66     int num1 = 5, num2 = 10;
67
68     resultBool = (num1 == num2);
69     Console.WriteLine($"Equal (==): {resultBool}"); // Output: False
70
71     resultBool = (num1 > num2);
72     Console.WriteLine($"Greater Than (>): {resultBool}"); // Output: False
73
74     resultBool = (num1 < num2);
75     Console.WriteLine($"Less Than (<): {resultBool}"); // Output: True
76
77     resultBool = (num1 >= num2);
78     Console.WriteLine($"Greater Than or Equal (>=): {resultBool}"); // Output:
False
79
80     resultBool = (num1 <= num2);
81     Console.WriteLine($"Less Than or Equal (<=): {resultBool}"); // Output: True
82
83     resultBool = (num1 != num2);
84     Console.WriteLine($"Not Equal (!=): {resultBool}"); // Output: True
85
86     Console.WriteLine();
87     #endregion
88
89     #region 04. LOGICAL OPERATORS
90     Console.WriteLine("04. Logical Operators:");
91
92     bool lx = true, ly = false, lz;
93
94     lz = lx && ly;
95     Console.WriteLine($"AND (true && false): {lz}"); // Output: False
96
97     lz = lx || ly;
98     Console.WriteLine($"OR (true || false): {lz}"); // Output: True
99
```

```
100         lz = !lx;
101         Console.WriteLine($"NOT (!true): {lz}"); // Output: False
102
103         Console.WriteLine();
104         #endregion
105
106         #region 05. BITWISE OPERATORS
107         Console.WriteLine("05. Bitwise Operators:");
108
109         int a = 12;    // 00001100
110         int b = 25;    // 00011001
111
112         result = a & b; // 00001000 = 8
113         Console.WriteLine($"Bitwise AND (12 & 25): {result}"); // Output: 8
114
115         result = a | b; // 00011101 = 29
116         Console.WriteLine($"Bitwise OR (12 | 25): {result}"); // Output: 29
117
118         result = a ^ b; // 00010101 = 21
119         Console.WriteLine($"Bitwise XOR (12 ^ 25): {result}"); // Output: 21
120
121         Console.WriteLine();
122         #endregion
123
124         #region 06. INCREMENT/DECREMENT OPERATORS
125         Console.WriteLine("06. Increment/Decrement Operators:");
126
127         // Post-Increment
128         x = 10;
129         int postInc = x++;
130         Console.WriteLine($"Post-Increment: x = {x}, result = {postInc}"); // Output:
x = 11, result = 10
131
132         // Pre-Increment
133         int y = 10;
134         int preInc = ++y;
135         Console.WriteLine($"Pre-Increment: y = {y}, result = {preInc}"); // Output: y
= 11, result = 11
136
137         // Post-Decrement
138         x = 10;
139         int postDec = x--;
140         Console.WriteLine($"Post-Decrement: x = {x}, result = {postDec}"); // Output:
x = 9, result = 10
141
142         // Pre-Decrement
143         y = 10;
144         int preDec = --y;
145         Console.WriteLine($"Pre-Decrement: y = {y}, result = {preDec}"); // Output: y
= 9, result = 9
146
147         Console.WriteLine();
148         #endregion
149
```

```
150      #region 07. TERNARY OPERATOR
151      Console.WriteLine("07. Ternary Operator:");
152
153      a = 20;
154      b = 10;
155      result = (a > b) ? a : b;
156      Console.WriteLine($"Ternary Result (a > b ? a : b): {result}"); // Output: 20
157
158      Console.WriteLine();
159      #endregion
160
161      Console.ReadKey();
162    }
163  }
164 }
165
```