Programming Fundamentals Lab

Lab # 05

Data Types, Logical, Relational, Increment, Decrement and sizeof Operator

Instructor: Fariba Laiq

Email: fariba.laiq@nu.edu.pk

Course Code: CL1002

Semester Fall 2022

Department of Computer Science,

National University of Computer and Emerging Sciences FAST Peshawar Campus

Contents

Data Types:

In C programming, data types are declarations for variables. This determines the type and size of data associated with variables. Today we will discuss 3 data types: int, bool, float.

Type	Size (bytes)	Format Specifier
int	at least 2, usually 4	%d, %i
float	4	%f
bool	1	%d

Types of Logical Operators in C

We have three major logical operators in the C language:

- Logical NOT (!)
- Logical OR (||)
- Logical AND (&&)

Following table shows all the logical operators supported by C language. Assume variable **A** holds 1 and variable **B** holds 0, then −

Operator	Description	Example
&&	Called Logical AND operator. If both the operands are non-zero, then the condition becomes true.	(A && B) is false.
||	Called Logical OR Operator. If any of the two operands is non-zero, then the condition becomes true.	(A || B) is true.
!	Called Logical NOT Operator. It is used to reverse the logical state of its operand. If a condition is true, then Logical NOT operator will make it false.	!(A && B) is true.

Operator	precedence
!	High
&&	Medium
||	Low

Example Code:

```
#include <stdio.h>
int main()
{
    int a = 5, b = 5, c = 10, result;
    result = (a == b) && (c > b);
    printf("(a == b) && (c > b) is %d \n", result);
    result = (a == b) && (c < b);
    printf("(a == b) && (c < b) is %d \n", result);
    result = (a == b) || (c < b);
    printf("(a == b) || (c < b) is %d \n", result);
    result = (a != b) || (c < b);
    printf("(a != b) || (c < b) is %d \n", result);
    result = !(a != b);
    printf("!(a != b) is %d \n", result);
    result = !(a == b);
    printf("!(a == b) is %d \n", result);
    return 0;
}
```

Output:

```
a = 5  b = 5  c = 10
(a == b) && (c > b) is 1
(a == b) && (c < b) is 0
(a == b) || (c < b) is 1
(a != b) || (c < b) is 0
!(a != b) is 1
!(a == b) is 0
```

C Increment and Decrement Operators

C programming has two operators increment `++` and decrement `--` to change the value of an operand by 1.

Increment `++` increases the value by 1 whereas decrement `--` decreases the value by 1. These two operators are unary operators, meaning they only operate on a single operand.

Example Code:

```
#include <stdio.h>
int main()
{
    int a = 10, b = 100;
    float c = 10.5, d = 100.5;
    printf("++a = %d \n", ++a);
    printf("--b = %d \n", --b);
    printf("++c = %f \n", ++c);
    printf("--d = %f \n", --d);
    return 0;
}
```

Output:

```
Initial value of a = 10

++a = 11

Value of a now = 11

a++ = 11

Value of a now = 12

++a = 13
```

C Relational Operators

A relational operator checks the relationship between two operands. If the relation is true, it returns 1; if the relation is false, it returns value 0.

Relational operators are used in decision making and loops.

Operator	Meaning of Operator	Example
==	Equal to	5 == 3 is evaluated to 0
>	Greater than	5 > 3 is evaluated to 1
<	Less than	5 < 3 is evaluated to 0
!=	Not equal to	5 != 3 is evaluated to 1
>=	Greater than or equal to	5 >= 3 is evaluated to 1
<=	Less than or equal to	5 <= 3 is evaluated to 0

Example Code:

```
#include <stdio.h>
int main()
{
    int a = 5, b = 5, c = 10;
    printf("%d == %d is %d \n", a, b, a == b);
    printf("%d == %d is %d \n", a, c, a == c);
    printf("%d > %d is %d \n", a, b, a > b);
    printf("%d > %d is %d \n", a, c, a > c);
    printf("%d < %d is %d \n", a, b, a < b);
    printf("%d < %d is %d \n", a, c, a < c);
    printf("%d != %d is %d \n", a, b, a != b);
    printf("%d != %d is %d \n", a, c, a != c);
    printf("%d >= %d is %d \n", a, b, a >= b);
    printf("%d >= %d is %d \n", a, c, a >= c);
    printf("%d <= %d is %d \n", a, b, a <= b);
    printf("%d <= %d is %d \n", a, c, a <= c);
    return 0;
}
```

Output:

```
5 == 5 is 1

5 == 10 is 0

5 > 5 is 0

5 > 10 is 0

5 < 5 is 0

5 < 10 is 1

5 != 5 is 0

5 != 10 is 1

5 >= 5 is 1

5 >= 10 is 0

5 <= 5 is 1

5 <= 10 is 1
```

The sizeof operator

The sizeof is a unary operator that returns the size of data i-e how much memory it takes.

Example Code:

```
#include <stdio.h>
int main()
{
    int i;
    float f;
    char c;
    bool b;
    printf("Size of int=%d bytes\n",sizeof(i));
    printf("Size of float=%d bytes\n",sizeof(f));
    printf("Size of char=%d byte\n",sizeof(c));
    printf("Size of char=%d byte\n",sizeof(b));
    return 0;
}
```

Output:

```
Size of int=4 bytes
Size of float=4 bytes
Size of char=1 byte
Size of char=1 byte
```

References:

https://www.programiz.com/c-programming/c-operators

https://www.tutorialspoint.com/cprogramming/c_logical_operators.htm