CL1002 – Programming Fundamentals Lab

Lab # 04

Arithmetic Operators & Escape Sequences in C

Instructor: Muhammad Saood Sarwar

Email: saood.sarwar@nu.edu.pk

Department of Computer Science,

National University of Computer and Emerging Sciences FAST Peshawar

Escape Sequences

Character combinations consisting of a backslash (\) followed by a letter or by a combination of digits are called "escape sequences." To represent a newline character, single quotation mark, or certain other characters in a character constant, you must use escape sequences. An escape sequence is regarded as a single character and is therefore valid as a character constant. Escape sequences are used to format our output. The following escape sequences can be used to print out special characters.

Escape Sequence	**Description**
\n	Newline
\t	Horizontal tab
\v	Vertical tab
\\	Backslash
\"	Double quote

To insert a line break, a new-line character shall be inserted at the exact position the line should be broken. In C, a new-line character can be specified as \n (i.e., a backslash character followed by a lowercase n).

Example 1

```
#include <stdio.h>

int main() {
    printf("This is a line of text.\n");  // \n represents a newline
    printf("This is a new line of text.\n");  // \n creates a new line

    return 0;
}
```

Example 2

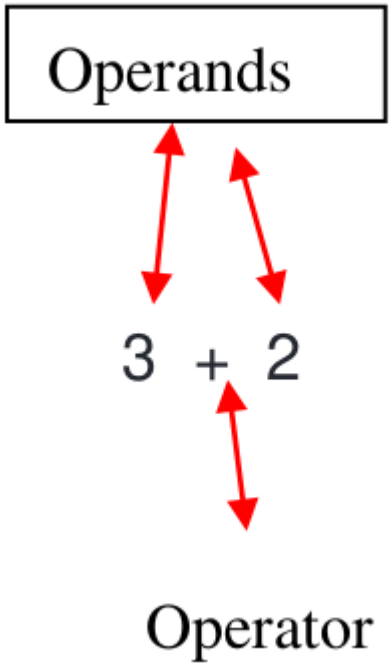
The following program shows the use of Newline Escape Sequence (\t).

```
#include <stdio.h>

int main() {
    printf("Name:\tJohn Doe\n");  // \t represents a tab
    printf("Age:\t30\n");  // \t inserts a tab to align the columns

    return 0;
}
```

Now try escape sequences \\ , **\v** , \" yourself.

Operators

Operators are special symbols that carry out arithmetic or logical computation. The value that the operator operates on is called the operand.



Here, + is the operator that performs addition. 2 and 3 are the operands and 5 is the output of the operation.

```
int sum1 = 100 + 50;      // 150 (100 + 50)
int sum2 = sum1 + 250;    // 400 (150 + 250)
int sum3 = sum2 + sum2;   // 800 (400 + 400)
```

Arithmetic Operators

Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication etc.

Operator	Meaning	Example
+	Add two operands	x + y
-	Subtract right operand from the left	x - y
*	Multiply two operands	x * y
/	Divide left operand by the right one	x / y
%	Modulus - remainder of the division of left operand by the right	x % y (remainder of x/y)

Assignment & Assignment Arithmetic operators

Assignment operators are used to assign values to variables.

int a = 5 is a simple assignment operator that assigns the value 5 on the right to the variable as on the left.

Operator	Example	Equivalent to
=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5

Example 3

```
#include<stdio.h>
int main()
{
   int a=2;
   int b=3;
   int sum=a+b;
   int diff=a-b;
   int product=a*b;
   int division=a/b;
```

```
    int mod=a%b;
    printf("a = %d b = %d", a, b);
    printf("\nSum: %d", sum);
    printf("\nDiff: %d", diff);
    printf("\nProduct: %d", product);
    printf("\nDivision: %d", division);
    printf("\nModulus: %d", mod);
return 0;
}
```

Example 4

```
#include <stdio.h>

int main() {
    // Declare and initialize variables
    int a = 10;
    int b = 5;

    // Assignment arithmetic operators
    a += b; // Equivalent to a = a + b
    printf("a += b: %d\n", a);

    a -= b; // Equivalent to a = a - b
    printf("a -= b: %d\n", a);

    a *= b; // Equivalent to a = a * b
    printf("a *= b: %d\n", a);

    a /= b; // Equivalent to a = a / b
    printf("a /= b: %d\n", a);

    a %= b; // Equivalent to a = a % b
    printf("a %%= b: %d\n", a);

    return 0;}
```

Exercise:

1. **Practice the examples provided above for better understanding. (4 examples)**
2. **Write a code in C, that generates the following output using escape sequences.**

   Character 1: "Hello there!"

   Character 2: 'Hi! How are you? Character 1: I'm good, thanks!

   Character 2: That's great!' Character 1: \\Bye\\
3. **Find the value of y from the following equation. Read value of x from user.**

   $y= 3x^3 - 2x^2 - x + 2$
4. **Write a C program that simulates a bank account transaction.** The program should take three inputs:

   The current account balance (represent this as **accountBalance** in your code) in dollars.

   The amount of money you want to deposit and withdraw (represent this as **depositTransactionAmount** and **withdrawTransactionAmount** in your code) in dollars.

   Perform the following operations using assignment arithmetic operators:

   Add the **depositTransactionAmount** to the **accountBalance** to simulate a deposit and store the result in **accountBalance**.

   Subtract the **withdrawTransactionAmount** from the **accountBalance** to simulate a withdrawal and store the result in **accountBalance**.

   After each operation, display the updated account balance (**accountBalance**).

   Provide the code for this program and a sample output with user inputs.