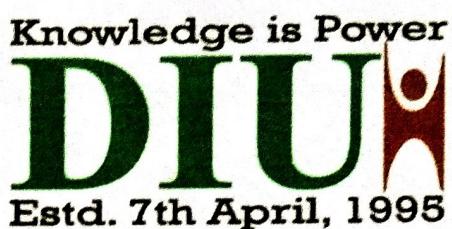


# Dhaka International University



## DEPARTMENT OF CSE

### LAB REPORT

**COURSE NAME** : Structured Programming Language Lab

**COURSE CODE** : 613 - 102

**REPORT NO** : 04

**REPORT ON** : Using round(), log(), log10(), exp() and cos() function in C programming

<u>SUBMITTED BY</u>	<u>SUBMITTED TO</u>
<p><b>NAME</b> : Mir Yeasin Abrar</p> <p><b>ROLL</b> : 35</p> <p><b>REG. NO</b> : CS-D-98-23-127358</p> <p><b>BATCH</b> : 98</p> <p><b>SEMESTER</b> : 1st</p>	<p>Rakib Mahmud Lecturer Department of CSE</p>

**DATE OF SUBMISSION** : 07 September 2024

Date of Performance: 31 August 2024

■ Lab Report : Using `round()`, `log()`, `log10()`, `exp()`, and `cos()` Functions in C Programming.

■ Objective :

1. Understand how to round floating-point numbers using '`round()`' function.
2. Learn how to calculate natural logarithms using '`log()`' function.
3. Explore the calculation of base-10 logarithms with '`log10()`' function.
4. Demonstrate the use of '`exp()`' function for computing exponential values.
5. Use the '`cos()`' function to find the cosine of an angle in radians.

■ Introduction : This report explores five fundamental mathematical functions from the C '`math.h`' library.

- `round()` : This function rounds a floating point number to the nearest integer.
  - `round(2.5)` yields  $3.0$
  - `round(-2.3)` yields  $-2.0$
- `log()` : This function computes the natural logarithm (base  $e$ )
  - $\log(2.718)$  approximates  $1.00$ , since  $\log_e(e) = 1$

- $\log_{10}()$ : This function calculates the base-10 logarithm.  
-  $\log_{10}(1000.0)$  yields 3.0, as  $10^3 = 1000$
- $\exp()$ : computes the exponential function  $e^x$  where e is Euler's number.  
-  $\exp(1)$  yields approximately 2.72, since  $e^1 = e$
- $\cos()$ : This function computes the cosine of an angle in radians.  
-  $\cos(0.0)$  yields 1.0, as  $\cos(0) = 1$

**Discussion :** In this section we will learn about the 'Presentation of code', 'Output', and Explanation of the code. The shown code will be as in 'IDE' and the output will be as in console.

... code:

```
#include <stdio.h>
#include <math.h>

int main()
{
    // Rounding
    double num1 = 2.5;
    double num2 = -2.3;
    printf("Round of num1 : %.1lf\n", round(num1));
    printf("Round of num2 : %.1lf\n", round(num2));
```

// Logarithms

```
double n = 2.718;  
double m = 1000.0;
```

```
printf("Natural log of n : %.1f\n", log(n));  
printf("Log10 of m : %.1f\n", log10(m));
```

// Exponential

```
double exp_val = 1.0;
```

```
printf("Exp of %.1f : %.3f\n", exp_val, exp(exp_val));
```

// cosine

```
double angle = 0.0; // angle in radian
```

```
printf("Cosine of '0' radian : %.1f\n", cos(angle));
```

```
return 0;
```

```
}
```

... output :

Round of num1 : 2.5

Round of num2 : -2.0

Natural log of n : 1.0

Log10 of m : 3.0

Exp of 1.0 : 2.714

Cosine of '0' radian : 1.0

...Explanation:

#include <stdio.h> - for input/output operations and  
#include <math.h> for mathematical functions like  
round(), log(), log10(), exp(), cos().

variables:

- For rounding num1, num2
- For logarithms n, m
- For exponential exp\_val
- For cosine angle

All the variables are double type and also returns double type value. Here %.1lf / %.3lf represents double type format specifier.

Function call:

- printf() - prints output in the console.
- round(num1) - round(2.5) in 3.0
- round(num2) - round(-2.3) in -2.0
- log(n) - Natural log (2.718) into 1.0
- log10(m) - Lo base log (1000.0) into 3.0
- exp(exp\_val)-exponential of  $e^1$  in 2.718
- cos(angle) - cosine of 0 radian into 1.0

return 0 - returns to main function.

**Conclusion:** In this section we will discuss about the challenges we encountered as 'limitations' and the use of these functions as 'applications'.

### ... Limitations :

1. Floating point Precision : Floating-point operation may introduce precision errors due to the limited representation of decimal numbers.
2. Domain Restrictions : Functions like  $\log()$  and  $\log_{10}()$  require positive inputs; otherwise, they return 'NaN' (not a number)
3. Angle Units : The  $\cos()$  function expects the angles in radians. Degrees must be converted to radians for accurate results.

### ... Applications :

1. `round()` - useful in rounding financial figures and for precise data analysis.
2.  $\log()$  and  $\log_{10}()$  : crucial for scientific computation and applications requiring logarithmic scales.
3. `exp()` : Applied in scenarios involving exponential growth or decay, such as population modeling.
4. `cos()` : Important in fields like trigonometry, physics, and engineering for analyzing periodic phenomena.

## References:

- C Standard Library Documentation
- Teach yourself C by Herbert Schildt
- learn.microsoft.com
- linkedin.com/learning
- github.com
- openai.com
- freeCodeCamp.org
- giraffeacademy.com