

Assignment 1-A

Q1. Write a C program to calculate the area of a rectangle using user input for length and width. Use appropriate arithmetic operators and expressions.

Ans:

```
#include <stdio.h>
int main() {
    float len, wid, area;
    printf("Enter the length of the rectangle: ");
    scanf("%f", &len);
    printf("Enter the width of the rectangle: ");
    scanf("%f", &wid);

    area = len * wid;

    printf("Area of the rectangle is: %.2f\n", area);
    return 0;
}
```

Q2. Explore the associativity of various operators in C programming by creating a program that involves multiple operators with different associativity. Explain the results.

Ans:

```
#include <stdio.h>

int main() {
    int result;
    result = 10 - 5 * 2;
    printf("Result 1: %d\n", result);
    result = (10 - 5) * 2;
    printf("Result 2: %d\n", result);
    int a, b;
    a = b = 5;
    printf("Result 3: a = %d, b = %d\n", a, b);
    result = 8 & 12 | 2; //bit-wise operator
    printf("Result 4: %d\n", result);
    int x = 5, y = 10;
    result = (x > y) ? x : y; //ternary operator
    printf("Result 5: %d\n", result);
    return 0;
}
```

1. Example 1 demonstrates how parentheses can change the precedence of operations. Without parentheses, multiplication has higher precedence than subtraction, but with parentheses, subtraction is performed first.

2. Example 2 shows the right-to-left associativity of assignment operators. Here, `b` is assigned the value 5 first, then `a` is assigned the value of `b`.

3. Example 3 illustrates the precedence and associativity of bitwise operators. Bitwise AND (`&`) has higher precedence than bitwise OR (`|`), so `8 & 12` is calculated first, then `2 | (8 & 12)`.

4. Example 4 showcases the associativity of the ternary operator `?:`, which is right-to-left. `(x > y) ? x : y` is evaluated first, and depending on the condition, either `x` or `y` is selected.