**Computer Programming [IIITS]**
**First Semester 2020-2021**
**Lab-3 (24/12/2020)**

---

Goals for the lab:

1. Learn about basic data types: variables, data types and sizes, operators, expressions, precedence and associativity.

2. Experiment with arithmetic operations involving integers and floating point numbers.

Dealing with different types of data:

- **char:** The most basic data type in C. It stores a single character and requires a single byte of memory in almost all compilers.

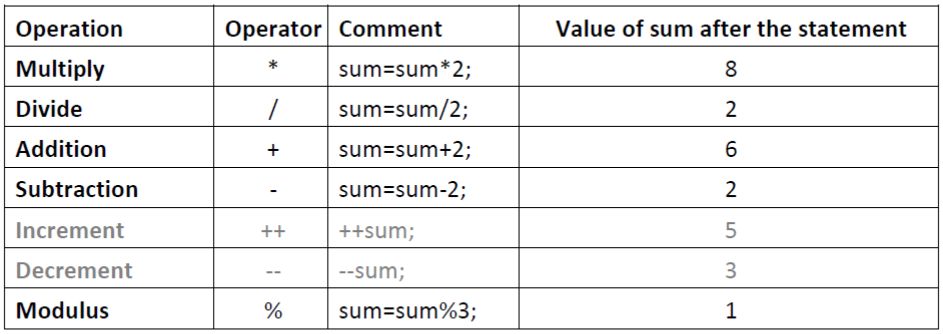
- **int:** It is used to store an integer value.

- **float:** It is used to store decimal numbers (numbers with floating point value) with single precision.

- **double:** It is used to store decimal numbers (numbers with floating point value) with double precision.

| Data Type | Format Specifier | Data Type | Format Specifier |
|---|---|---|---|
| short int | %hd | unsigned short int | %hu |
| unsigned int | %u | int | %d |
| long int | %ld | unsigned long int | %lu |
| long long int | %lld | unsigned long long int | %llu |
| signed char | %c | unsigned char | %c |
| float | %f | double | %lf |
| | | long double | %Lf |

Arithmetic operators: The symbols of the arithmetic operators and their usage are given below. Initial value of sum is defined as "int sum=4;"



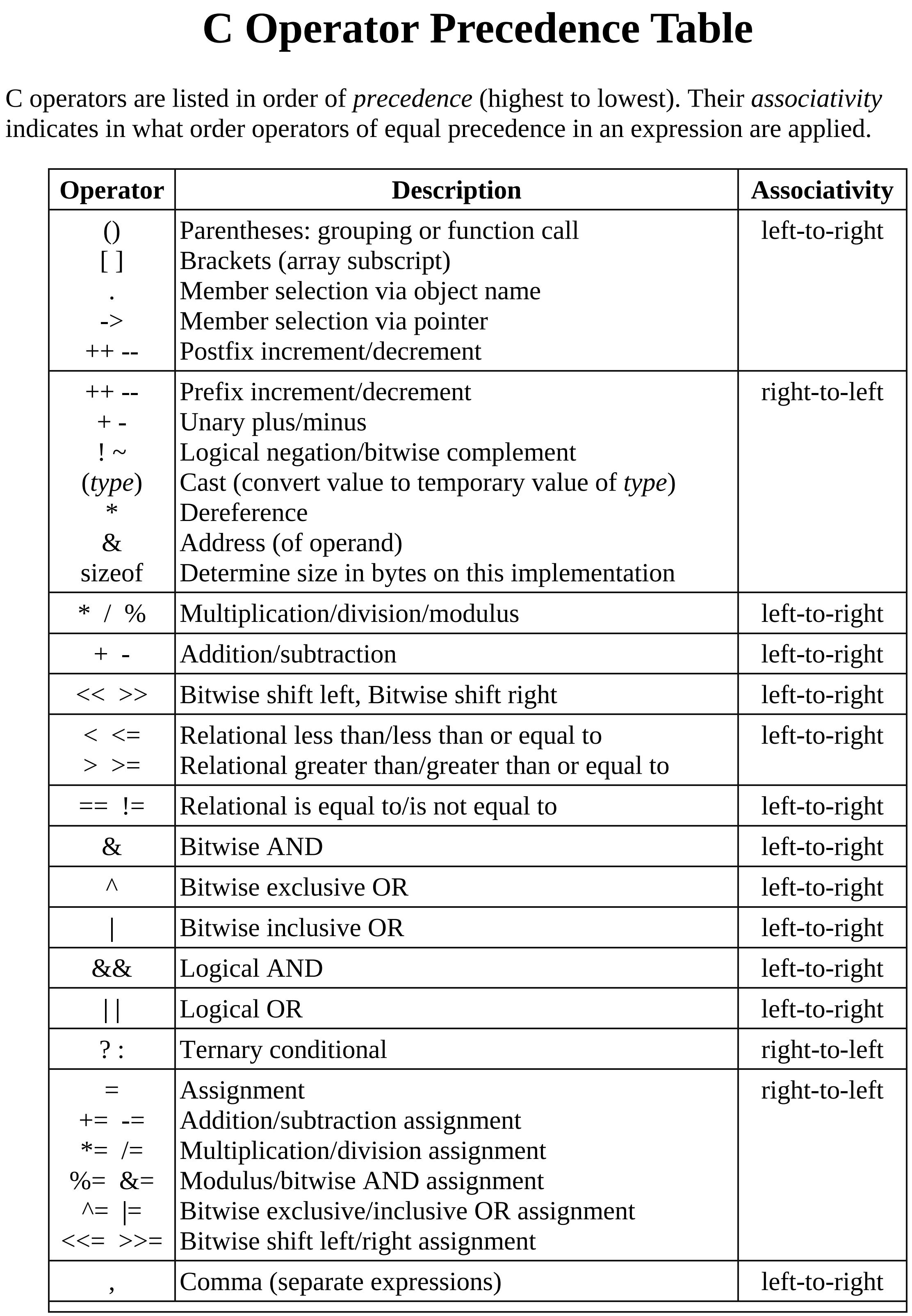
| **Operation** | **Operator** | **Comment** | **Value of sum after the statement** |
|---|---|---|---|
| **Multiply** | * | sum=sum*2; | 8 |
| **Divide** | / | sum=sum/2; | 2 |
| **Addition** | + | sum=sum+2; | 6 |
| **Subtraction** | - | sum=sum-2; | 2 |
| Increment | ++ | ++sum; | 5 |
| Decrement | -- | --sum; | 3 |
| **Modulus** | % | sum=sum%3; | 1 |

Example: Compile and run the following programs.

```c
#include <stdio.h>
int main()
{
   int i=2, j=3, k, l;
   float a ,b;
   k=i/j*j;
   l=j/i*i;
   a=i/j*j;
   b=j/i*i;
   printf(‘‘%d %d %f %f\n”,k,l,a,b);
return 0;
}
```

```c
#include<stdio.h>
main( )
{
   int a,b;
   printf(‘‘Give side of square”);
   scanf(‘‘%d”,&a); //reading integer
   b=a*a;
   printf(‘‘The area is %d”,b);
}
```

1. Write program, which reads as input sides of a rectangle and prints its area.
2. Write program, which reads 4 numbers a, b, c and p. Let $f(x) = ax^2 + bx + c$ be a function. The program outputs the value of $f(p)$. e.g. input 4 3 -1 2, output $4(2)^2 + 3(2) - 1 = 21$.
3. Program to find the sum of last two digits.
   input 13613, output $1 + 3 = 4$ and input 324, output $2 + 4 = 6$.
4. Write program to print the second last digit. input 23617, output 1.
5. Read two numbers. Find their product after exchanging last digits. Input 4270 and 153 output 640950 ($4273 \times 150$). Input 348 and 31 output 12958 ($341 \times 38$).



# C Operator Precedence Table

C operators are listed in order of *precedence* (highest to lowest). Their *associativity* indicates in what order operators of equal precedence in an expression are applied.

| Operator | Description | Associativity |
|---|---|---|
| ()<br>[ ]<br>.<br>-><br>++ -- | Parentheses: grouping or function call<br>Brackets (array subscript)<br>Member selection via object name<br>Member selection via pointer<br>Postfix increment/decrement | left-to-right |
| ++ --<br>+ -<br>! ~<br>(*type*)<br>*<br>&<br>sizeof | Prefix increment/decrement<br>Unary plus/minus<br>Logical negation/bitwise complement<br>Cast (convert value to temporary value of *type*)<br>Dereference<br>Address (of operand)<br>Determine size in bytes on this implementation | right-to-left |
| * / % | Multiplication/division/modulus | left-to-right |
| + - | Addition/subtraction | left-to-right |
| << >> | Bitwise shift left, Bitwise shift right | left-to-right |
| < <=<br>> >= | Relational less than/less than or equal to<br>Relational greater than/greater than or equal to | left-to-right |
| == != | Relational is equal to/is not equal to | left-to-right |
| & | Bitwise AND | left-to-right |
| ^ | Bitwise exclusive OR | left-to-right |
| \| | Bitwise inclusive OR | left-to-right |
| && | Logical AND | left-to-right |
| \|\| | Logical OR | left-to-right |
| ? : | Ternary conditional | right-to-left |
| =<br>+= -=<br>*= /=<br>%= &=<br>^= \|=<br><<= >>= | Assignment<br>Addition/subtraction assignment<br>Multiplication/division assignment<br>Modulus/bitwise AND assignment<br>Bitwise exclusive/inclusive OR assignment<br>Bitwise shift left/right assignment | right-to-left |
| , | Comma (separate expressions) | left-to-right |