

# **ED1021 - Introduction to computation and visualisation**

## **L5 - Arithmetic Expression**

**Ramanathan Muthuganapathy (<https://ed.iitm.ac.in/~raman>)**

**Course web page: <https://ed.iitm.ac.in/~raman/introcomp.html>**

**Moodle page: Available at <https://courses.iitm.ac.in/>**

# Arithmetic operators

## Done using well-known symbols

- + (Addition)
- - (Subtraction)
- \* (Multiplication)
- / (Division) - Note down the symbol
- % (Reminder)

# Arithmetic operators

## Types

- Unary - operates on only one variable / constant e.g.  $-a$ ,  $-5$
- Binary - operates on only two variables / constants (or a mix of both) e.g.  $a+b$ ,  $c-3$ ,  $d*e$ ,  $f/g$ ,  $h+2.34$
- Ternary - we will see this later!

# Arithmetic Expression

## Syntax

- combination of variables, constants and operators. Some examples
  - $a - b$
  - $a + b - c * d / e,$
  - $c \% d - 5$
- Qn: Give some examples for what is NOT an expression?

**Qn: Suppose  $a = 5$ ,  $b = 4$ ,  $c = 2$ , what are the possible answers for the expression  $a + b / c$  ?**

# Operator precedence

**-a + b \* c / d % e**

- Evaluates the exp from left to right
- Unary minus
- %, /, \* - whichever happens from left to right
- +, -

# Some common mistakes

**Arithmetic expressions should not be written in a typical math. way**

- $2 * a$  in math exp is  $2a$  (but what will happen if you use  $2a$  in the A. E.?)
- $a / b$  should not be written as ....

# Use of parentheses

- suppose you have the A.E.  $a + b / a - b$ , write the equivalent math. exp.
-



**CW: 1) Write the A.E. for a quadratic equation?**

# Assignment statement uses arithmetic expression etc.

- Take the statement  $c = a + b$
- You add the two variables 'a' and 'b' and put the resultant onto another variable called 'c' (Qn: How did you represent this in a flow chart?)
- NOTE: Operator precedence for assignment is from right to left.

# Putting all together - variable, constants, A.E., assignment

## Writing a source code in C.

```
#include <stdio.h> // Header file - mandatory
int main( )
{
    // Declaration of variables - int, float etc.
    // Arithmetic exp. and assignment
    // output statement using printf( )
}
```

IMP: 1) Every variable should be declared as a specific data type (int, float etc.) before used.

2) Every statement should end with the symbol ; (semicolon)

If you use the symbol ; then it is taken as the end of that statement.

# Example program

## example1.c

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

```
int main( )  
{  
    int a, b, c;  
    a = 5; b = 10;  
    c = a + b;  
    printf(“%d \n”, c);  
}
```

# Building the source code

## General procedure

- Create the project as discussed in the video.
- add the file source code as a new item
- Use build or compile
- Check the output for error (see the error window - should show 1 succeeded 0 failed)
- Run the code

NOTE: Program works line by line, sequentially (does not jump, unless said so!)

CW: (1) Create the project and add the source code and show the output?  
(2) Modify the code to do other operations such as -, \*, and /. What are the observations?  
(3) Modify the code with declaring the variables as 'float'. Use %f instead of %d in the printf statement.

# Name the operators

# Operator precedence



# Integer division

## important one to note

consider the expression  $i / 10 * 10$ , what will be the value if this exp. if  $i$  is 45?

Will you get the same value for the exp:  $i * 10 / 10$

- NOTE: When there is an integer subexpression with  $/$ , integer division will yield only the truncated values.
- What will be the value of the following expression:  $1 / 2 * 50$  ?

# How to get correct answer?

- When there is a mix of float and int, the integer will be elevated to float locally
- Elevating even one of them to a float. e.g.  $i / 10.0 * 10$
- Use type casting - locally elevate the subexpression to float.
  - $(\text{float})i / 10 * 10$
- NOTE: the variable  $i$  acts as float only in that subexpression!

# Demo using L5\_floatint.c

# Assignment expression

## Arithmetic expression and Assignment statement

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

```
int main( )
```

```
{
```

```
    int a, b, c;
```

```
    a = 5;
```

```
    printf("%d \n", a);
```

```
    a = a + 1;
```

```
    printf("%d \n", a);
```

```
    a += 1; // Assignment expression
```

```
    printf("%d \n", a);
```

```
}
```

# More examples

- You can use Assign. Exp. for any operator (almost!)
- E.g.  $a -= b + c$ ,  $c *= d / e$ ,  $f \% = g$
- In general  $\langle \text{var} \rangle \langle \text{op} \rangle = \langle \text{expr} \rangle$

# Unary increment and decrement

- Instead of using  $x = x + 1$ , you can say  $x++$  or  $++x$
- $x++$  is called post-increment
- $++x$  is called pre-increment

# How do they work?

```
int x, y;  
x = 5;  
y = x++; // y = x, and then x = x+1  
printf("%d %d\n", x, y);
```

```
int x, y;  
x = 5;  
y = ++x; // x = x+1 and then y = x  
printf("%d %d\n", x, y);
```

**CW: Update the operator  
precedence.**



# Giving value to a variable - input

- // Assigning value

```
int a;  
float b;  
a = 2;  
b = 4.75;
```

- // Defining a var

```
int a = 2;  
float b = 4.75;
```

- // Using scanf

```
int a;  
float b;  
scanf("%d %f", &a, &b);
```

NOTE: The symbol & is very important.

In VS2019, scanf\_s has to be used.

# Demo using L5\_input.c

**CW: WAP to convert a given  
fahrenheit to centigrade?**

**HW: Read chapter 7 on input and output and test them using programs.**