



Arab Academy for Science and Technology and Maritime Transport

College of Engineering and Technology

Computer Engineering Department

CC112 Structured Programming

Lecture 4

LECTURE 4

Arithmetic operations

LECTURE OUTLINE

- i. **Arithmetic Operators**
 - Shortcut assignment
 - Prefix form
 - Postfix form

- ii. **Arithmetic Expression**
 - Precedence rules
 - Evaluate the expression

- iii. **Math in C**
 - Math library
 - Math library examples

I. ARITHMETIC OPERATORS

- Most C programs perform calculations using the C arithmetic operators
- Note the use of various special symbols not used in algebra.
- In algebra, to multiply a times b , we simply place these single-letter variable names side by side as in ab .
- In C, however, if we were to do this, ab would be interpreted as a single, two-letter name (or identifier).
- Therefore, C requires that multiplication be explicitly denoted by using the $*$ operator as in $a*b$.

I. ARITHMETIC OPERATORS

C operation	Arithmetic operator	Algebraic expression	C expression
Addition	+	$f + 7$	<code>f + 7</code>
Subtraction	-	$p - c$	<code>p - c</code>
Multiplication	*	bm	<code>b * m</code>
Division	/	x / y or $\frac{x}{y}$ or $x \div y$	<code>x / y</code>
Remainder	%	$r \text{ mod } s$	<code>r % s</code>

I. ARITHMETIC OPERATORS

Division and the Remainder Operator

- Arithmetic expressions in C must be written in straight-line form to facilitate entering programs into the computer.
- Thus, expressions such as $\frac{a}{b}$ “a divided by b” must be written as a/b so that all operators and operands appear in a straight line.
- Integer division yields an integer result and float division yields a float result.
- For example:
if both values are integers then $7 / 4 = 1$
if any of the values are float then $7 / 4 = 1.75$

I. ARITHMETIC OPERATORS

Division and the Remainder Operator

- C provides the remainder (modulus) operator, %, which yields the remainder after integer division.
- The remainder operator is an integer operator that can be used only with integer operands.
- The expression $x \% y$ yields the remainder after x is divided by y .
- Thus, $7 \% 4$ yields 3 and $17 \% 5$ yields 2.

I. ARITHMETIC OPERATORS

Shortcut assignment

“Short cut” assignment operators combine an operation with an assignment.

<code>a += b</code>	<code>a = a + b</code>
<code>a -= b</code>	<code>a = a - b</code>
<code>a *= b</code>	<code>a = a * b</code>
<code>a /= b</code>	<code>a = a / b</code>
<code>a %= b</code>	<code>a = a % b</code>

For instance, instead of writing:

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

you could write

```
a += 1;
```


I. ARITHMETIC OPERATORS

Prefix form

- Prefix increment and decrement operators increment or decrement the variable, then return its resulting value.

```
int a, b;  
a = b = 10;  
printf("%d\n", ++a);    /* Prints 11 */  
printf("%d\n", a);     /* Prints 11 */  
printf("%d\n", --b);   /* Prints 9 */  
printf("%d\n", b);     /* Prints 9 */
```

- Remember: If the `++` comes *before* the variable, it increments *before* determining the result.

I. ARITHMETIC OPERATORS

Postfix form

- Postfix increment and decrement operators return the original value of the variable, then increment or decrement the variable.

```
int a, b;  
a = b = 10;  
printf("%d\n", a++);      /* Prints 10 */  
printf("%d\n", a);       /* Prints 11 */  
printf("%d\n", b--);     /* Prints 10 */  
printf("%d\n", b);       /* Prints 9  */
```

II. ARITHMETIC EXPRESSION

- **An expression is a valid arrangement of variables, constants, and operators.**
- **In C each expression can be evaluated to compute a value of a given type**

II. ARITHMETIC EXPRESSION

Rules of Operator Precedence

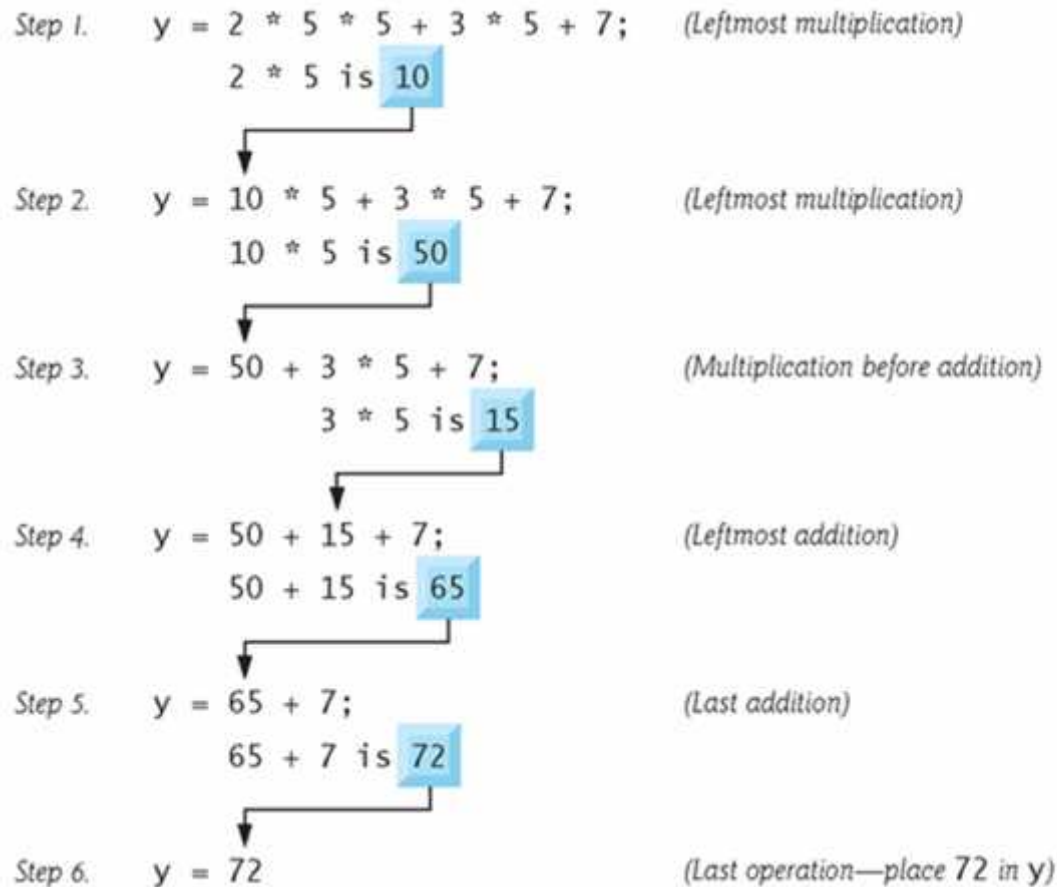
- C applies the operators in arithmetic expressions in a precise sequence determined by the following rules of operator precedence, which are generally the same as those in algebra:

Operator(s)	Operation(s)	Order of evaluation (precedence)
()	Parentheses	Evaluated first. If the parentheses are nested, the expression in the <i>innermost</i> pair is evaluated first. If there are several pairs of parentheses "on the same level" (i.e., not nested), they're evaluated left to right.
*	Multiplication	Evaluated second. If there are several, they're evaluated left to right.
/	Division	
%	Remainder	
+	Addition	Evaluated third. If there are several, they're evaluated left to right.
-	Subtraction	
=	Assignment	Evaluated last.

II. ARITHMETIC EXPRESSION

Rules of Operator Precedence

EXAMPLE:



II. MATH IN C

Math library

- The C math library provides a lot of useful predefined math functions.
- A function is a subprogram used to do a certain task. A function has zero or more inputs (called parameters), and zero or one output (called return value)



II. MATH IN C

Math library

- Before you use them, remember to include the math library in your code:

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

- function sqrt:

```
y = sqrt ( x );
```

II. MATH IN C

Math library

sin(x) cos(x) tan(x)

sqrt(x) pow(x,y) abs(x)

log(x) log10(x) exp(x)

fabs(x) floor(x) ceil(x)

II. MATH IN C

Example 1:

Write a program to get the roots of a quadratic equation, given the 3 coefficients a, b, and c,

$$a x^2 + b x + c = 0$$

$$\text{Root}_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad \text{Root}_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

Solution:

```
disc = pow(b,2) - 4 * a * c;
```

```
root_1 = (-b + sqrt(disc)) / (2 * a);
```

```
root_2 = (-b - sqrt(disc)) / (2 * a);
```

THANK YOU