Mathematical expressions

- **Division operator and modulus operator**
  - The result of **division operator /** depends on the type of its operands
  - If one or both operands has a floating point type, the result is a floating point type.
  - If both operands are integer type, the result is an integer

  Examples: 11/4 =2 11.0/4.0 =2.75 11/4.0=2.75
  How about 15/4 ? 19/3 ? 15.0/4? 15.0/4.0?

  - **Modulus operator (%)** is used **with integer type operands** and always has an integer type result
  - The result of modulus operator is the remainder of the corresponding integer division operation

  Examples: 11%4=3 5%8 =5
  How about 12 % 5? 20%3 ?

  Application examples:
  - random turn for robot wander behavior (east, west, south, north)
  - if there are a total of 135 students. Each classroom can have a maximum of 25 students, how many classroom is needed? If the classroom is always filled to maximum capacity before the next classroom is used for assignment, how many students is in the last classroom?
  - For a total of 3645 seconds, how many minutes and seconds is it equivalent to?

- **Operator precedence:**

Determines which operator is applied first in an expression with several operators

Precedence	operator	description
higher	( )	function call
	-	negation (unary), right associative
	*, /, %	multiplication, division, modulus
	+, -	addition, subtraction
lower	=	assignment

In C++, binary operators having the same precedence level are left associative
Example: -7 * 10 – 6 % 3 * 4 + 9 / 2

How about: 5%4*5 + 10/3*5 – 10

Parenthesis can be used to change the usual order. Parts in () are evaluated first.
Example: (7 * (10 - 5) % 3) * (4 + 9) / 2

- **Type coercion and type cast in assignment statements**

  Type coercion is automatic conversion of an operand to another type

  Hierarchy of Types

  Highest: `long double`
  `double`
  `float`
  `unsigned long`
  `long`
  `unsigned int`
  Lowest: `int`
  Ranked by largest number they can hold

  Type Coercion rules:
  1) char, short, unsigned short automatically promoted to int
  2) When operating on values of different data types, the lower one is promoted to the type of the higher one.
  3) When using the = operator, the type of expression on right will be converted to type of variable on left

  Example:
  ```
  float someFloat;
  someFloat = 12;
  ```
  ```
  int someInt;
  someInt = 4.8;
  ```

  Example:
  ```
  char c='a';
  int x = 3;
  x = x + c;
  ```
  ```
  float someFloat = 2.5;
  int someInt = 3;
  cout << someFloat+someInt << endl;
  ```

  Type casting is explicit conversion of type
  Examples: static_cast<int>(4.6) static_cast<float>(5)

  static_cast<float>(7/4) static_cast<float>(7)/4

  What values are stored in "loCost" and "hiCost" after the following statements are executed?
  ```
  float  loCost=12.342, hiCost=12.348;

  loCost = (static_cast<int> (loCost * 100.0 + 0.5) ) / 100.0;
  hiCost  = (static_cast<int> (hiCost * 100.0 + 0.5) ) / 100.0;
  ```

- **Function calls in expression**
  - **Function call** -- one function calls another by using the name of the called function together with () containing an argument list

    FunctionName (Argument list)

    Function call temporarily transfers control from the calling function to the called function. When the function's code has finished executing, control is transferred back to the calling block.

Argument list is a way for functions to communicate with each other by passing information. Argument list can contain 0, 1, or more arguments, separated by commas, depending on the function.

- **Where are functions? C++ standard library or written by programmers (user defined)**

- **Frequently used C++ standard library functions**

header file	function	example	value
<cstdlib>	abs(i)	abs(-6)	
<cmath>	fabs(x)	fabs(-6.7)	
<cmath>	pow(x, y)	pow(2.0, 3.0)	
<cmath>	sqrt(x)	sqrt(100.0)	
		sqrt(2.0)	
<cmath>	log(x)	log(2.0)	
<cmath>	sin(x)	sin(0.7853) // 45 degrees in radian	
	cos(x)	cos(0.7853)	
	tan(x)	tan(0.7853)	
<cmath>	floor(x)	floor(3.4)	
<cmath>	ceil(x)	ceil(-2.5)	

Practice question:

- Modulus operation is defined by Donald Knuth using floored division as this:
  For a%n → remainder = a – n*floor(a/n)

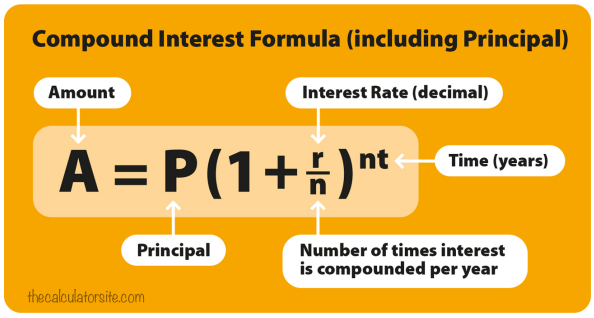
  For 5%3, remainder =5 – 3*floor(5/3)=2
  For -35%3, remainder= -35 – 3*floor(-35/3)= -35 – 3*(-11) -35+33 = -2

- Write the C++ program to compute the two roots of a quadratic equation : $ax^2+bx+c = 0$. For now, we assume that the equation always has two real roots.

  The program should prompt the user to enter the values of a, b, and c, and computes and displays the roots to the user

- Write a C++ program to compute the Euclidean distance between two points on a plane.
  The program should prompt the user to enter the co-ordinates of the two points, compute and display the Euclidean distance between the two points.

- Given the principle investment, the interest rate, the number of times in a year the interest is compounded, and the number of years of investment, compute the total anticipated future value of the investment at the end of the years allotted.



(refer to : https://www.thecalculatorsite.com/articles/finance/compound-interest-formula.php)

- The above problem can be easily transformed into computing the total payment on a loan, assuming we are given the initial loan amount, the loan interest rate, the number of times in a year the interest is compounded, and the number of years of the loan.