**Documentation in the Wiki**

The proposed section "Calculation functions" of the page <https://docs.moodle.org/35/en/Grade_calculations>  is:

 ==Calculation functions==

Every calculation must start with an equal sign (=). Following is an expression using operators and functions supported by the system.  
All common arithmetic operators are supported:

* addition, using the plus (+) sign
* subtraction, using the minus (-) sign
* multiplication, using the asterisk (\*) character
* division, using the slash (/) character
* exponentiation, using the caret (^) character

with their usual precedence rules: exponentiations are evaluated first, then multiplications and divisions are performed, finally additions and subtraction are carried out; so, the expression =1+2-3\*4/5^6 gives almost 3 (2,999232). Precedences can be forced using parentheses, as in the expression =((((1+2)-3)\*4)/5)^6 which yields 0.

Comparison operators are supported:

* greater, using (>) character
* greater or equal, usign (>=) characters
* less, using (<) character,
* less than, using (<=) characters
* equal, using (==) characters

each comparison operator evaluates to 0 when false and 1 when true. They can be used in any expression with precedence over the rest of the operators. I.e. the expression =2<3\*10 yields 10 because 2<3 yields 1, but for legibility it is recommended to use parentheses or the 'if' function as in: =(2<3)\*10 or =if(2<3, 10, 0) that are quantitatively equivalent.

**Note**: Moodle does not allow calculations involving no ID numbers.

Functions can also appear in expressions, using the comma (,) character to separate their arguments listed within function (round) brackets. (The separator character could be a semicolon (;) in other [languages](https://docs.moodle.org/dev/index.php?title=Language&action=edit&redlink=1), see below).

* average([[item1]], [[item2]]...): Returns the average of the values in a list of arguments
* max([[item1]], [[item2]]...): Returns the maximum value in a list of arguments
* min([[item1]], [[item2]]...): Returns the minimum value in a list of arguments
* mod(dividend, divisor): Calculates the remainder of a division
* pi(): Returns the value of the number Pi
* power(base, power): Raises a number to the power of another
* round(number, count): Rounds number to count decimal digits
* floor(number): Maps a real number to the largest previous integer
* ceil(number): Maps a real number to the smallest following integer
* sum([[item1]], [[item2]]...): Returns the sum of all arguments
* if([[item1]], [[item2]], [[item3]]): Evaluates the first argument (condition) and returns the second argument if the condition is not zero (true condition) and returns the third argument if the condition is zero (false condition).
* or([[item1]], [[item1]]…): Further Condition/ Needs to be used with the first argument (condition) of the if-clause. Evaluates whether one of the following arguments (conditions) within the or-clause is true and returns the true or false condition within the if-clause.  
  You can also use it in an encapsulated if-function with multiple or- and and-clauses.
* and([[item1]], [[item1]]…): Further Condition/ Needs to be used with the first argument (condition) of the if-clause. Evaluates whether all of the following arguments (conditions) within the and-clause are true and returns the true or false condition within the if-clause.  
  You can also use it in an encapsulated if-function with multiple or- and and-clauses.

Many other mathematical functions are also supported:

* sin()
* sinh()
* arcsin()
* asin()
* arcsinh()
* asinh()
* cos()
* cosh()
* arccos()
* acos()
* arccosh()
* acosh()
* tan()
* tanh()
* arctan()
* atan()
* arctanh()
* atanh()
* sqrt()
* abs()
* ln()
* log()
* exp()

**Example calculations**

* =max([[Quiz.1]], [[Quiz.4]], [[Assignment.1]]) - Returns the maximum value of Quiz.1, Quiz.4 and Assignment.1
* =average(max([[Quiz.1]], [[Quiz.4]], [[Assignment.1]]), min([[Quiz.1]], [[Quiz.4]], [[Assignment.1]])) - Returns the average of the maximum and the minimum values among Quiz.1, Quiz.4 and Assignment.1 (functions can be nested)
* =sum([[1]]\*0.3, [[2]]\*0.6, [[3]]\*2) - Returns a weighted grade calculations where item 1 is weighted 30%, item 2 is weighted at 60% and item 3 is weighted at 200%
* =if([[midtermexam]]>=5, [[midtermexam]]+[[lab]], 0) - Returns the sum of midtermexam and lab items if the midtermexam grade is 5 of more, and 0 otherwise. This is a very handy way of dealing with conditional evaluation in a course.
* =if(or(sum([[exercise1]],[[exercise2]],[[exercise3]])>15, [[finalexam]]<50), 0, 1) - Returns 0 if one of the following arguments is true: the sum of the exercises is less than 15 points OR the value of the finalexam is less than 50 points in total, and 1 otherwise. This is an additional way to combine two or more existing functions or conditions.
* =if(and(sum([[exercise1]],[[exercise2]],[[exercise3]])>15, [[finalexam]]<50), 0, 1) - Returns 0 if all of the following arguments are true: the sum of the exercises is less than 15 points AND the value of the finalexam is less than 50 points, and 1 otherwise. This is another way to combine two or more existing functions or conditions.