Introduction to Mathematical Operations in C

Since all information handled by a computer is numerical, it is important to have an understanding of how the computer performs operations on this information, and what limitations there are in these operations. Some of the limitations in how operations are handled will be more readily explained by recalled the earlier work in this course with binary numbers.

Basic Operators

Operation	Operator in C	Comments
Addition	+	Works in the same manner as in 'regular' math. Must consider overflow if the result exceeds the maximum size for that data type.
Subtraction	-	As above. Integers give integer results, floats give a float result.
Multiplication	*	Can NOT use an 'x' or simply put bracket around values. i.e. (2)(3+4) will NOT be understood.
Division	/	The result depends on the data types. Division of integers will give a truncated integer result; Division involving float values will give a float result.
Modulo	%	Gives the remainder of an integer division. For example, 34%10 = 4

Casting

The C language has the useful feature built in which can be used to force (cast) data from one type to another. For example, a float value can be cast into an integer as shown below.

```
#include <stdio.h>

int main(void)
{
        int answer;
        float aNum = 123.4;
        answer = (int)aNum;                    //This casts aNum to an integer
        return 0;
}
```

Exercises

1 Complete the table below for each expression, and then check your answer by calculating the expression in a simple C program and outputting with a printf(). The first on is done for you!

EXPRESSION	Predicted Value & type	Checked in C program
3*4	12 int	Correct
5.0 – 3.5		
9/3		
9/3.0		
45 + 5.5		
2*(3+4)		
3(7-5)		
123/10		
123%10		
(3*4+5)%2		

2. Write a C program called cMath.c that has two integer variables called num1 and num2 and performs all of the mathematical operations on these two values. Output the results on separate lines using separate printf() statements. For division, have the value output both as an integer result and as the correct decimal result. A sample of what your output should look like is shown below.

The first number is 21. The second number is 5.

21 + 5 = 26
21 – 5 = 16
21 * 5 = 105
21/5 = 4 (as integer)
21/5 = 4.2 (as decimal)
21%5 = 1

Program By: Ima Programmer. September 10, 2017