Variables in C

The C language provides the four basic arithmetic type specifiers char, int, float and double, and the modifiers signed, unsigned, short and long. Variables are signed unless otherwise specified. On a PC, this results in the following variables:

Type	Bytes	Bits	Range
char	1	8	-128 -> 127
unsigned char	1	8	0 -> 255
short int	2	16	- 32768 -> +32767
unsigned short int	2	16	0 -> +65535
int	4	32	–2147483648 -> 2147483647
unsigned int	4	32	0 -> +4,294,967,295
long int	4	32	–2147483648 -> 2147483647
unsigned long	4	32	0 -> 4294967295
long long int	8	64	-2^{63} -> $2^{63}-1$
unsigned long long	8	64	0 -> $2^{64}-1$
float	4	32	Properties unspecified
double	8	64	Properties unspecified
long double	12	96	Properties unspecified

Type qualifiers:
const – used on constants that are not changed
volatile – variables that can be changed between different accesses, e.g. in an interrupt. Can also be used to avoid optimization of the variable for debugging.

Fixed width integers, defined in C99
Signed: int8_t, int16_t, int32_t, int64_t
Unsigned: uint8_t, uint16_t, uint32_t, uint64_t

Data in program-memory (Flash/programspace)
http://www.nongnu.org/avr-libc/user-manual/pgmspace.html

Global and local variables.
A variable declared inside a function is called a local variable. It’s only available inside the function. Variables are not automatically initialized and should be so before being used. Value/contents are forgotten between each call of the function, but static can be used for remembering a variable between each call of the function:
static uint8_t demo_var;
Static variables are initialized before first run of a function.

A variable declared outside of all functions (before main) is called a global variable. It can be accessed by every function in the file scope. A program can have same name for local and global variables but the value of local variable inside a function will take preference.

Local variables can be declared where they are used, not necessarily in the beginning of the function. There is no need to re-use variables as the compiler optimizes storage automatically. Local variables are better optimized by the compiler.

Operators and priority in C
https://en.wikipedia.org/wiki/Operators_in_C_and_C%2B%2B

Overflow
In compiler.
In calculations.

Strings - pointers

Calculation time

Float takes time as there is no HW support.

Memory occupation
Including floating point takes

Fixed point calculations
Multiply by factor of 2. (left shift – fast)
Add for rounding
Divide by factor of 2 for result (right shift – fast).