

Types, Operators, and Expressions

① Variable Names

- * letters and digits
- * starts with letter or $_$ - (library routines use $_$ to start var names)
- * case-sensitive.
- * lower case
- * UPPER CASE - Symbolic Constants.
- * Keywords - if, else, while, etc. can't be used.
- * name your variables wisely!

② Data Types and Sizes

char
int
float
double

} only 4 basic data types in C!

short
long

} qualifiers.

eg.

short int sh;
long int counter;

(or)

short sh;
long counter;

* Size of data types on "most" 32-bit machines

char - 1 byte
short - 2 bytes
int - 2 or 4 bytes
long - 4 bytes.

* The compiler is free to choose appropriate sizes for its own

Only Restriction:

$\text{short} \leq \text{int} \leq \text{long}$ (HW hardware)
at least 16 bits at least 16 bits at least 32 bits

Signed or Unsigned (Qualifiers)

applied to: char! or ints.

unsigned numbers - obey modulo 2^n arithmetic.

char (8 bits)

signed

-128 to 127

-2^7 to 2^7-1

(2's complement machine)

unsigned

0 to 2^8-1
(255)

#code1
SHOW CODE IN CLASS?

#Homework

default qualifier for char

machine-dependant

HW 1 - Find the qualifier for plain char in your machine

long - can also be applied to double.
⇒ long double.

(*) headers → $\langle \text{limits.h} \rangle$ and $\langle \text{float.h} \rangle$

eg. INT_MAX +32767
INT_MIN -32767

UINT_MAX 65535.

#HW2 - Find the values for the following constants.
CHAR_MAX, CHAR_MIN, LONG_MAX, LONG_MIN,
SHRT_MAX, SHRT_MIN.

3. Constants

1234 - int

u or U - unsigned constant

123456789l } - long
(or)
123456789L }

eg. 707u

ul or UL - unsigned long const.

eg. ~~2302 2303 2362~~ too long
2302 2362 UL

Floating point constants
123.45 or 1e-2

default - double

f or F - float | l or L - long double

eg. 123.4f 123.4567L

(4)

Octal and hexadecimal

eg. $31_{10} = 037 = 0X1F$
 ↓ zero (octal) ↓ zero X (hexadecimal)

$0XFUL = 15_{10}$ (unsigned long).

Character constant

'x' → value is ASCII value of x.

eg. '0' = 48 | 'a' = 97 | 'A' = 65
 zero in single quotes

* Character constants participate in numeric operations just as any other integers.

#code 2 - show code for counting the characters in a string of chars (line of text).

Escape Sequences

\n - newline

\t - horizontal tab

'\0' ⇒ character with value zero (null character) (X)
 '\0' = 0 ('\0' is preferred)

Constant expression

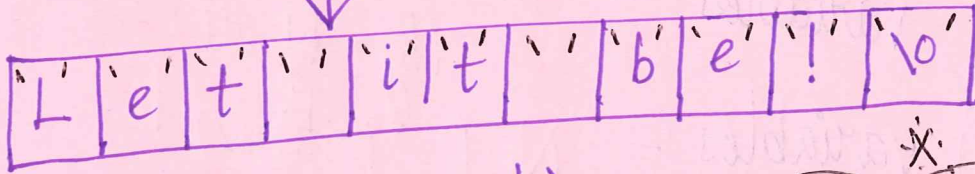
(5)

```
#define MAXLINE 1000
char line[MAXLINE + 1];
```

const expr.

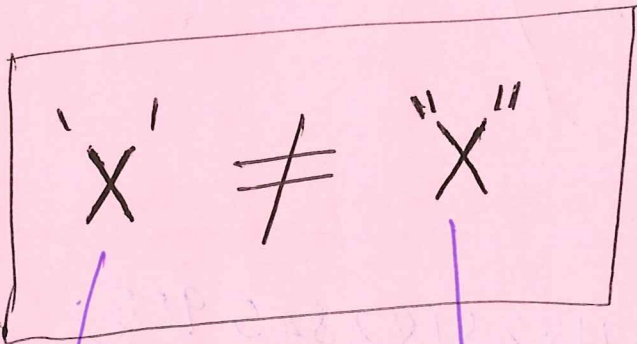
String constant or string literal

"Let it be!" or "" (empty string)



strlen - <string.h>

* #code 3 - code for strlen



∴ 'X' = 88 (ascii value of X)

"X" = ['X', '\0']

integer used to represent x in machine's char set.

array of chars.

(alternatives to #define)

```
Enumeration constant
enum boolean { NO, YES };
              ↓   ↓
              0   1
```

```
enum months { JAN = 1, FEB,
              MAR, APR,
              NOV, DEC };
```

④ Declarations

⑥

```
int lower;  
int integers[100];  
int upper = 1000;  
char c;  
char line[1000];
```

Initialization (X)

External or static variables - initialized to zero (by default)

Automatic (local) variables - NOT initialized.

contains **GARBAGE VALUE!**

#code 4 - show this in class.

"const" qualifier

```
const double e = 2.71828182845905;
```

```
const char msg[] = "warning: ";
```