$\boxed{\text{Integers}}$

Types

- We will tell you the # bytes for each integer type that's implementation defined
  - short, int, long, size_t
- Char is always one byte
  - The signedness is implementation defined for char
- short, int, long are all signed
- size_t is unsigned

Conversion / Signedness

- Suppose our integers have k bits
  - Recall 1 byte = 8 bits = 2 hex digits
- binary to decimal

$$\underset{\pm 2^{k-1}}{\underline{1}} \quad \underset{2^{k-2}}{\underline{1}} \quad \underset{2^{k-3}}{\underline{0}} \quad \cdots\cdots \quad \underset{2^{2}}{\underline{0}} \quad \underset{2}{\underline{1}} \quad \underset{2^{0}}{\underline{0}}$$

$\uparrow$ sign bit

- Signed: The value at left is -2^{k-1}
- unsigned: The value at left is $+2^{k-1}$

- If the integer is signed & sign bit is 1, the value is negative
- binary to hex
  - Take 4 bits at a time (starting from right side) and convert

bin	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
hex	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
dec	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

  - Append an "0x" to the left
- Note the sign bit is the leftmost bit in k-bit representation, NOT what you see on paper

  Ex: int x = 0xFF // "leftmost bit" is 0, not 1

Range of Values

- integer w/ k bits, then the possible values it can take on are:
  - Signed: $[-2^{k-1}, 2^{k-1}-1]$ Ex: int: 32 bits $[-2^{31}, 2^{31}-1]$
  - unsigned: $[0, 2^{k}-1]$ Ex: unsigned int: 32 bits $[0, 2^{32}-1]$

Undefined Behavior

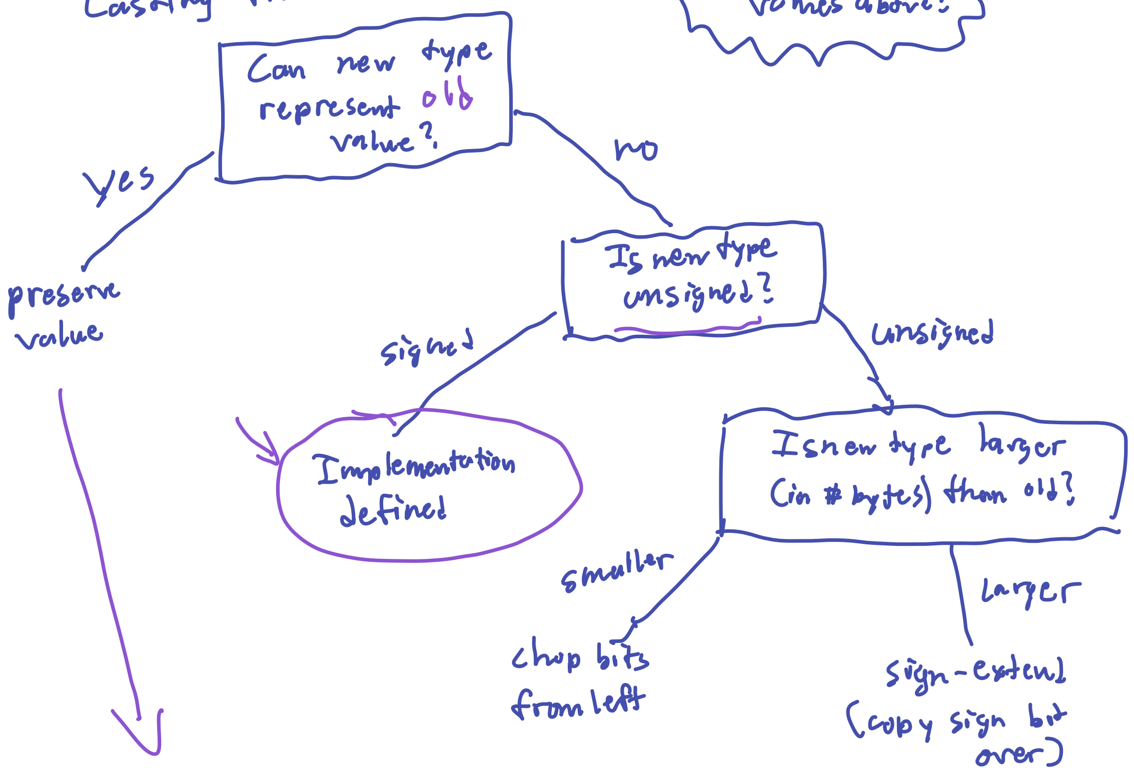
- overflow on signed integers is UB (-fwrapv is on) unless
- overflow on unsigned integers is NOT UB
- bit shifts by k negative or by equal or larger # bits is UB
  x << k

Casting Rules

- Suppose you had t1 x = ______
  t2 y = (t2) x;

Casting from t1 to t2

Use the range of values above!

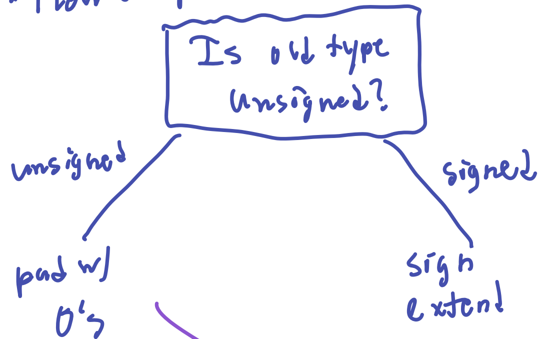


0111 . . . 1111$_2$

short x = 0x 7FFF

int y =

- How to preserve value?



Ex: unsigned short x = 0x FFFF → 0x 0000 FFFF

- Integer constants count as type int

  Ex: 32 is type int

  - unless specified (in which casting occurs)

  Ex: short x = 32; → type short

- If you mix & match types, implicit casting occurs

```
int x = ____;
size_t y = ____;
y = x; // Implicit casting (y = (size_t)x;)
    if(x < y); // Implicit casting! Don't worry about
               what happens, though. Always be explicit
```

Ex: Assume short 2 bytes, int 4 bytes,

```
short a = 0xDD; // What's the value?
signed char a0 = (signed char)a;
unsigned char a1 = (unsigned char) a;
```

Soln:

value: 035

Impl. defined!

hex: 0xDD val: 221

```
short b = -3;
int b0 = (int) b;
unsigned char b1 = (unsigned char) b;
```

hex: 0x FF FD

hex: 0xFFFF FFFD val: -3

hex: 0xFD val: 253

```
size_t y = -1;
```

hex: 0x FFFF FFFF FFFF FFFF

val: 1.84×10^{19}

☆ What happens? If not implementation-defined, what is value?
what is hex rep?