



Principles of Computer Science II

Nadeem Abdul Hamid
CSC121A - Spring 2005

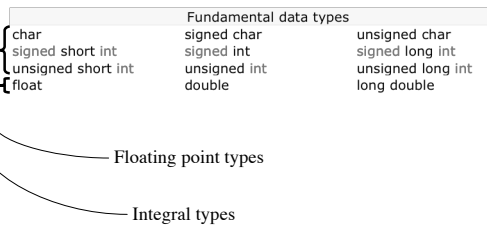
Lecture Slides 13 - Data Types

Types

- Purpose of types for variables and constants
 - Tell compiler appropriate amount of space to reserve in memory
 - Allow compiler to use proper machine instructions to carry out operations
- Expressions (made up of constants, variables, fn calls) also have a *value* and *type*

2

Basic C Data Types



3

Characters

- Any integral type can be used to represent a character
 - Constants such as 'a' and '+' are of type int not char
- Each char is stored in one byte of memory (usually 8 bits)
- At the bit level: `char c = 'a';`

```

0 1 1 0 0 0 0 1

```
- One byte (8 bits) can store 256 distinct values
- Type char is equivalent to either signed char or unsigned char
 - Signed char range: -128 ... 127
 - Unsigned char range: 0 ... 255

4

Data Type int

- The principal working type of C
- The default type of integers worked with on a machine
- Typically...
 - 2 bytes (16 bits) on personal computers, or
 - 4 bytes (32 bits) on high-end workstations/mainframes
- In 2 bytes the range is: -32768 ... 32767
- In 4 bytes the range is: -2147483648 ... 21477483647
- * Be careful about integer overflow in programs
- Besides decimal integer constants, also
 - Hexadecimal: `0xa1`
 - Octal: `0377`
- Note: `11 != 011` /* with leading zero is octal constant */

5

Types short, long, unsigned

- Intended for specialized use
- Storage is a concern: use short
 - Compiler *may* provide less storage for a short than an int (not required to do so)
- Large integer values needed: use long
- Typically
 - short = 2 bytes
 - long = 4 bytes
- unsigned: to store integer values without a sign

6

Types of Constants

- On 2-byte int machine, compiler treats
 - 32000 as int
 - 33000 as long
- Programmer can append suffixes to specify types of integer constants
 - 37u or 37U - unsigned
 - 37l or 37L - long
 - 37ul or 37UL - unsigned long

7

Floating Point Types

- C's default floating type is `double`
 - Usually, `float` stored in 4 bytes
 - About 6 decimal places of accuracy
 - `double` stored in 8 bytes
 - About 15 decimal places of accuracy
- Precision: number of significant decimal places
- Range: limits of largest and smallest values
- Example: `float` precision is about 6 and range is approx. 10^{-38} to 10^{+38}
 - $0.d_1d_2d_3d_4d_5d_6 \times 10^n$ /* -38 <= n <= 38 */
- `double` precision is about 15 places and range is approx. 10^{-308} to 10^{+308}

8

Floating Point on the Computer

- Not all real numbers are representable using "floating point" types
- Floating point arithmetic (unlike integer arithmetic) may not be exact
 - For large computations, especially rounding effects, etc. need to be taken into account (numerical analysis)

9

sizeof Operator

- Looks like a function, but is an *operator*
 - `sizeof a + b == sizeof(a) + b`
- Returns an integer (usually unsigned) representing number of bytes needed to store the object (or an object of that type) in memory
- Guarantees
 - `sizeof(char) == 1`
 - `sizeof(short) <= sizeof(int) <= sizeof(long)`
 - `sizeof(signed) == sizeof(unsigned) == sizeof(int)`
 - `sizeof(float) <= sizeof(double) <= sizeof(long double)`
- Sizeof program...
- See `float.h` and `limits.h` (pg.528) for some predefined constants

10

Look over in textbook

- Section 6.8 (Mathematical functions)
- Section 6.9 (Conversions and casts)
- 6.12 (Common programming errors)
- 6.13 (System considerations)

11

Enumeration Types (Ch. 7)

- User-defined types to name a finite set of elements (essentially represented as integers)
- Enumerators (elements) can be initialized explicitly
 - Repeated values ok, but identifiers must be unique
- Variables can be declared at the same time as the enum declaration
- In general, used as programmer-specified constants
 - Can be converted to int representation with a cast

12

Defining “Synonyms”: typedef

```
typedef int color;  
color red, blue, green;
```

➤ A common use is with enumeration types

13