Faculty of Computer Science, Dalhousie University *3-Oct-2018*

CSCI 2132 — Software Development

Lecture 13: C Basic Types

Location: Chemistry 125 Instructor: Vlado Keselj
Time: 12:35 – 13:25

Previous Lecture

- C operators expressions and statements
- Operators, expressions: Java vs. C
- C control structures; comparison with Java
- Implementation-defined and unspecified behaviour
- Logic expressions
- Differences between C99 and earlier standards
- goto statement, null statement

Slide notes:

More about Integer Size

- C does not specify size of integers
- Unlike Java, which requires 32 bits for an integer
- Reason: Efficiency in using directly hardware architecture
- To get the range of the current machine use:
  `#include <limits.h>`
- Defines some constants, such as `INT_MIN` and `INT_MAX`
- A note about long constants: we can use suffix L; e.g., `15L`

The C standard does not specify what is the exact size of the `int` type. As a comparison, Java does require an `int` type to be 32 bits long. The C does not specify this for efficiency reasons. Different computers with different CPU units generally work with groups of bits of different length. This basic group of bits is called a *machine word* and it has generally different length on different computers. For example, a 64-bit computer will have a word lenght of 64 bits. Since C has flexibility of definining an `int` to be of different size, the compiler can always choose to store an `int` into one machine word to maximize program efficiency.

What about the sizes of short and long? C does not specify the sizes of short/long, either. Some rules are: int cannot be shorter than short int, and long int cannot be shorter than int. This flexibility again lets designers of C compilers to come up with the most efficient implementation for a specific platform.

How do programmers determine the range of int then? There are some macros defined in the standard C library. If you include `<limits.h>` in your program (i.e., use `#include <limits.h>`), then you can use the following two macros as the minimum and maximum values that can be stored in an int variable on your platform:

```
INT_MIN
INT_MAX
```

There are similar macros defined for long, short, etc. in the same header file.

In C integer constants are considered to be of the type int by default, if they are not out of the range of int. If they are, a type with a larger range (such as long int) will be used. The rules regarding which types should be used are different in C89 and C99. If you are interested, read the textbook for details. For the compiler to treat a constant as long, we can explicitly suffix the constant with either L or l (the character l, not digit 1). An example: 15L

Floating-Point Number Types

There are three floating types in C: float for single-precision floating-point numbers; double for double-precision floating-point numbers and long double for extended-precision floating-point numbers. How much precision each type provides is implementation-defined. Most modern compilers choose to follow the IEEE standard 754. There are more details in the textbook, but in most cases, it is enough to know that float keeps 6 significant digits while double keeps 15 significant digits.

Floating constants are double by default.

Character Types

Slide notes:

Character Types

- `char` is character type, stored in 8 bits, or 1 byte
  - unlike Java, which uses 16 bits
- Example:
  ```
  char ch = 'a';
  ```
- C treats characters as small integers (usually signed integers)
- For example, take a look at the following code:
  ```
  if ('a' <= ch && ch <= 'z')
    ch = ch - 'a' + 'A';
  ```
- What does it do?