APS105 Lecture 2
1.1-1.4

Dan Zingaro

January 20, 2010

Feedback from Reading Quiz

- "Won't we waste memory if we use a `long` to store small numbers like 32?" — Yes!
- `printf` is the function in `stdio.h`, not `print`
- "What's that `void` on the function `main`?
  - It means that `main` takes no information "from the outside"; the operating system just tells it to start
  - We'll see later that we can replace `void` with other stuff that allows the operating system to pass information to your program

Why C?

- Computers understand only simple instructions like “add these two numbers” or “tell me if this number is 0”
- These instructions are called machine language
- Computers do not understand C
- C is a high-level language that has to be translated to machine language before the computer can run it
- A compiler performs this “translation” for us

ConcepTest

Consider the following statements about compilers: (I) Compilers translate C code to machine language
(II) Compilers must know the type of machine on which the generated machine code will run
(III) Compilers detect all possible program errors
(IV) A compiler is not required if you write directly in machine language
Which of these statements are true?

- A. All statements are true
- B. (I) and (III) only
- C. (I), (II), and (IV) only
- D. (I), (II), and (III) only

Hello World Program

```
#include <stdio.h>

int main (void)
{
  // This program prints a simple message
  printf("Hello, world\n");
  return 0;
}
```

Hello World: Explanation

- `#include <stdio.h>`
  - C comes with a standard library of functions
  - We must tell the compiler which ones we intend to use in our code
  - Information about `printf` (and many other input/output functions) is found in file `stdio.h`
- `int main (void)`
  - A C program's execution starts in the function called `main`

Hello World: Explanation...

- `printf("Hello, world\n");`
  - `printf` is a function that outputs the stuff in the quotes to the screen
  - The `\n` means "advance to the next line"
  - `\n` is an example of what is called an escape sequence
  - Sometimes, we cannot type characters directly from the keyboard
  - In C, strings cannot span multiple lines, so we cannot just hit enter to make a new line!
  - Other escape sequences: `\\` (to print a backslash), `\"` (to print a quote)
- `return 0;`
  - This terminates the `main` function, and returns control to the operating system

ConcepTest

What is the output of the following `printf` function call?

```
printf ("I said, \"hi\"");
```

- A. `I said, "hi"`
- B. `I said, "hi`
- C. `I said, "hi""`
- D. `I said, "hi\"`
- E. Compilation error

Bits of Data

- The smallest piece of information in a computer is a bit
- A bit has value 0 or 1 (like a switch that must be off or on)
- One bit therefore has two possible states
- If we collect two bits together, how many total states do we have?
  - four states: 00, 01, 10, 11
- What about three bits?
  - Four states that start with 0: 000, 001, 010, 011
  - Four states that start with 1: 100, 101, 110, 111
  - Total: 8 states

ConcepTest

How many different states can we represent using five bits?

- A. 10
- B. 16
- C. 25
- D. 32

Integers

- One byte is 8 bits, so one byte has $2^8 = 256$ states
- To store integers, we could map the first state to integer 0, the second to integer 1, the third to integer 2 ...
- We'd also have to use some of the states for negative numbers
- Being able to store only 256 possible integers would not be very useful!
- In C, when we store integer values (using `int`), we're guaranteed that they use at least 16 bits
- If we want to store numbers of greater magnitude (positive or negative), we can use `long`, which guarantees a minimum of 32 bits
- `int` and `long` are integers: they are numbers that can be positive or negative, but cannot have a decimal portion
- To store numbers like 3.14, we use floating-point numbers instead