

CPSC 427: Object-Oriented Programming

Michael J. Fischer

Lecture 9
September 26, 2018

Following Specifications

Bytes and Characters

Overview of PS3

These abbreviated notes summarize lecture 9 given on September 26 but do not by any means fully capture what was presented.

Following Specifications

Why follow instructions?

A reasonable question is, “Why should I follow instructions when I know a different or better way of accomplishing the “same thing”?”

1. Programming is about producing code that fully satisfies design requirements.
2. If you don't like the requirements, it's reasonable to question them but not simply to ignore them.
3. For this course, the problem requirements also have a pedagogical purpose. When I say, for example, that a goal of the assignment is to learn how to use the C time functions from within C++, I mean exactly that. I'm not asking you to just figure out some way of determining the current year.
4. The ability to understand and follow instructions is a sign of maturity and professionalism that will help you in your career.

Bytes and Characters

History of ASCII

We had a long discussion of the history of character encodings, starting from 7-bit ASCII as used on early teletype machines up to current-day unicode.

Originally, the only characters that could be encoded on a computer were the ones that appeared on an English-language typewriter. There are so few such characters that they can be encoded in a single 8-bit byte.

At the time C was created, ASCII characters were all that were important to be able to read and write. Hence, type `char` became the name of a single-byte storage unit that could be used to represent a character (but could be used for other purposes as well).

Unicode

Unicode is a standard that assigns a unique numerical code to every letter and symbol in every language on earth. There are so many characters that the unicode encoding needs 32 bits.

These 32-bit quantities are usually themselves represented as sequences of one or more shorter storage units.

The commonly-used utf-8 encoding is a way of representing every unicode character by a sequence of one or more 8-bit bytes.

C/C++ works directly with bytes, not characters. A function like `in.get(ch)` reads a byte into `ch`, not a full character.

Note: The utf-8 encoding of every ASCII character is a single byte whose value is the same as its ASCII code.

Overview of PS3

Think-a-Dot

I gave an overview of the *Think-a-Dot* game. Everything I said is contained in the [PS3 assignment](#) and in some of the references cited there.