Design patterns are commonly-used architectural structures that help us organize systems. A pattern describes the core of a satisfactory solution to a problem that occurs over and over again. The study of design patterns originated in traditional architecture, and has been enthusiastically adopted by the software development community.

In software development, patterns have traditionally been structures of cooperating object-oriented (OO) classes in a software design. We are now expanding beyond this narrow limit to include large-scale software components, use cases and requirements, and the methodology of getting software built.

This talk will be a mile-wide, inch-deep survey of software patterns, starting with a review of Christopher Alexander’s bricks-and-mortar patterns and the classic 1995 “Gang of Four” OO design pattern catalog. From there, we’ll sample several emerging variants, including J2EE and .NET enterprise patterns, real-time patterns, and requirements patterns, as well as Jim Coplien’s organizational anti-anti-patterns.

Steve Masticola is a Senior Member of Technical Staff, software architect, and requirements engineer at Siemens Corporate Research. His current professional activities include global software development and “smoothing out” the transitions between stages of the software process. He holds a Ph.D. in computer science from Rutgers and is an IEEE Certified Software Development Professional and a Sun Certified Enterprise Architect. In his spare time, he and his wife enjoy hiking, geocaching, target shooting, motorcycle riding, and community service, though not necessarily all at once.

All ACM / IEEE-CS meetings are open to the public. Students and their parents are welcome. There is no admission charge, and refreshments are served.

A pre-meeting dinner with the speaker is held at 6:00 p.m. at Ruby Tuesday’s Restaurant on US 1. Please send email to princetonacm@acm.org in advance if you plan to attend the dinner.