both security and authentication of information. This course will begin with the problem of communicating between two computers, followed by the problem of building generalized networks for an arbitrary number of computers. Networking topics will include layered network structure, signaling methods, error detection and correction, flow control, routing, and protocol design and verification. We will then examine in detail a variety of encryption schemes, how they can be used, and how secure they are. Cryptographic topics will include classical cryptosystems, the data encryption standard, public-key cryptography, key escrow systems, and public policy on encryption. Offered in alternate years.

Requisite: COSC 112 or 201. Spring semester. Professor Kaplan.

301. Data Structures and Algorithms II. This course continues the exploration of data structures and algorithms that is begun in COSC 201. Topics include balanced search trees, amortized algorithms, graph data structures and algorithms, greedy algorithms, dynamic programming algorithms, NP completeness, and case studies in algorithm design.

Requisite: COSC 112 and 201. Fall semester. Professor L. McGeoch.

321. Computer Graphics. This course will explore the algorithms used to create "realistic" three-dimensional computer images. Major topics will include object representations (polygons, curved surfaces, functional models), rendering algorithms (perspective transformations, hidden-surface removal, reflectance and illumination, shadows, texturing), and implementation techniques (scan conversion, ray tracing, radiosity). Students will create images using OpenGl and Pixar's Renderman.

Requisite: COSC 112 or 201 or consent of the instructor. Omitted 2012-13.

341. Applied Algorithms. We will look at recent advances in the design and analysis of data structures and algorithms, with an emphasis on real-world applications. Topics to be covered include approximation algorithms and heuristics for NP-hard problems; combinatorial optimization; new analysis techniques; and methods of algorithm engineering and experimental analysis of algorithms. The specific problem domains to be studied will vary from year to year, to reflect the state of the art in algorithm research. Students will read and present research papers and carry out small research projects to evaluate algorithm performance in realistic scenarios.

Requisite: COSC 201. Omitted 2012-13.

371. Compiler Design. An introduction to the principles of the design of compilers, which are translators that convert programs from a source language to a target language. Some compilers take programs written in a general-purpose programming language, such as C, and produce equivalent assembly language programs. Other compilers handle specialized languages. For instance, text processors translate input text into low-level printing commands. This course examines techniques and principles that can be applied to the design of any compiler. Formal language theory (concerning regular sets and context-free grammars) is applied to solve the practical problem of analyzing source programs.

Topics include: lexical analysis, syntactic analysis (parsing), semantic analysis, translation, symbol tables, run-time environments, code generation, optimization, and error handling. Each student will design and implement a compiler for a small language. Offered in alternate years.

Requisite: COSC 112 and 161. Omitted 2012-13.