***Computer Engineering – Selected Software Track Course Descriptions:***

**These courses focus on both the strategy level of software architecture and design as well as the tactical level of implementation. Strategy is concerned with top-level structure, class relationships, data relationships, often represented using Universal Modeling Language (UML) diagrams. Design tactics are concerned with specific design principles and packaging techniques to support reusability and robustness of an implementation.**

1. [**CSE681**](file:///C:\su\WebPages\CSE681.xml) **- Software Modeling and Analysis (CSE Core Course for SW Track)**

**Software Architecture, strategy level design supported by UML modeling, threads, queues, distributed systems  
Midterm, three reports, two software projects.**

1. [**CSE686**](file:///C:\su\WebPages\CSE686.xml) **- Internet Programming**

**Client and Server side programming models, HTML, XSL, CSS, ASP, ADO  
In-class labs, one large final project to build a web site and web service.**

1. [**CSE687**](file:///C:\su\WebPages\CSE687.xml) **- Object Oriented Design (CSE Core Course for SW and HW Tracks)**

**C++ language, standard library, tactical level design and design principles  
Midterm, four challenging software projects.**

1. [**CSE775**](file:///C:\su\WebPages\CSE775.xml) **- Distributed Objects Processing**

**Win32 Systems programming, Component Object Technology, COM, ActiveX, .Net Controls  
Midterm, four software projects.**

1. [**CSE776**](file:///C:\su\WebPages\CSE776.xml) **- Design Patterns**

**Sophisticated, reflective view of Object Oriented Design at both the strategy and tactical levels,  
In-class presentations, small amount of software development.**

**te and present at the end of the course.**

1. [**CSE784**](file:///C:\su\WebPages\CSE784.xml) **- Software Engineering Studio**

**Learn to make critical evaluation of software products, e.g., specification, design, implementation, and test.  
Takehome Midterm - deep evaluation of one of your products.  
Case Study - class is partitioned into teams that specify, develop, and test a large distributed system.**

**All of the courses, with the exception of** [**CSE776 - Design Patterns**](file:///C:\su\WebPages\CSE776.xml)**, are project-based courses. Projects are modeled after industrial and commercial work, requiring careful design, robust implementation, and documentation of the source code you generate.**

***Computer Engineering Software Track Course Sequence:***

**I recommend that full time students in Computer Engineering - Software Track - take these courses in the following order:**

1. ***Fall:*  
   CSE681 - Software Modeling and Analysis (Core Course SW Track)  
   CSE661 - Advanced Computer Architecture (Core Course)  
   CIS554 - Object Oriented Programming with C++**
2. ***Spring:*  
   CSE687 - Object Oriented Design (Core Course SW and HW Tracks)  
   Elective course  
   CSE686 - Internet Programming**
3. ***Fall:*  
   CIS675 - Algorithms (Core Course)  
   CSE776 - Design Patterns (Prerequisite: CSE687)  
   CSE784 - Software Studio (Prerequisite: CSE681)**
4. ***Spring:*  
   CSE775 - Distributed Objects (Prerequisites: CSE681 and CSE687)  
   Master's Project**