

Advanced Website Programming

\$250.00

Continuing education only – not for college credit

Background: In general, "webmasters" can be categorized as either "form" or "substance". Form focuses on the presentation aspects of a web page, with the goal of maximum viewer impact. Substance focuses on the underlying infrastructure of a site, with emphasis on offering maximum functionality to the user while providing as much security against "crackers" (the proper name for criminally-inclined "hackers"). My experience in substantive website design and programming has been developed over the past 14 years (I currently maintain and regularly enhance 11 different "sites"), and has satisfied the high security standards set by a number of my high-tech clients.

Prerequisites: In order to cover the essential aspects of website security in a 1 week course, I will expect the students to possess a basic understanding of the open-source PHP programming language, at least a general familiarity with the open-source MySQL database query language, and, of course, a reasonably competent understanding of the standard web browser markup language, HTML. I will discuss the Cascading Style Sheet language, CSS, only peripherally, as this language is not essential to site structure, only page presentation. Note: the only one of these languages that may be truly considered as a "programming" language is PHP - none of the others is capable of performing even a simple mathematical or logical operation. In this regard, I shall provide a model browser-hosted "persistent connection" interface written in the open-source Javascript programming language, but I do not expect to have sufficient time to cover the interesting features either of my model module or of the language per se.

Course Description: The essential elements of site security I shall cover comprise:

1. Protecting code modules from exposure to outside the site by setting appropriate controls on the web server, of which I will use only the open-source Apache2.
2. Protecting site data base structures and content from exposure to outside attack via such common schemes as "SQL injection".
3. Restricting access to site data base(s) on a strictly-enforced user-based privilege scheme.
4. One of my special developments allows restriction to a strictly-defined sub-set of ALL site code modules on a user-based privilege scheme.

One other area I hope to cover is how, very efficiently, to support multiple logical sites (each with a different URL and, perhaps, a different sub-set of ALL available code "applications" and data base(s)), without depending on server "virtualization", which I happen NOT to trust. In my model infrastructure, ALL incoming queries are delivered DIRECTLY to my entry-portal "filter" module, where I decide which ones are legitimate and, if found to be so, I then direct those queries to the appropriate "application" handlers.

In the course of the course, I will offer many recommendations relating to coding standards that I have learned to be most effective over my 53 years of programming. In particular, I will encourage the students to consider the impact of their coding conventions on the number of machine cycles required to execute the code modules they develop. Thus, for example, I will NOT teach ANYTHING even vaguely resembling "templates" - they are ineffective crutches for sloppy and/or lazy programmers and I despise them! I WILL teach a modular site infrastructure that is quite efficient for both the programmer AND the server. I have little regard for a highly functional site if that functionality comes at the expense of excessively long response times. Using my techniques, I can create the basic functionality of an entirely new "application" in less than one programmer day ... and the response time is unlikely to be discernible from normal net round-trip delays by the normal user.

Jeff Meyers holds a J.D. Law, from Oklahoma City University Law School, a M.S. in Business Administration, specializing in Data Automation Processing, from George Washington University, and a B.S. in Mathematics, specializing in Computer Sciences, from the University of Oklahoma. Jeff was admitted to practice before the U.S. Patent and Trademark Office (1974), the Oklahoma State Bar (1976), and the Texas State Bar (1980). He is a member of IEEE and ACM. He has prepared and presented numerous papers at

continuing legal education forums on topics of interest to the IP Bar and has been 18 issued U.S. Patents in computer hardware and software, and more pending. He has professionally (and privately) prepared and executed many hundreds of computer programs, ranging in scale from quite modest to very large, using more than 35 different languages, many at the machine or assembly level and others at higher levels, on a wide spectrum of computers offered commercially by: International Business Machines: 1620, 1401/1410, System/360, *et al.*, Honeywell International, Inc.: H200/H800/H1800, Burroughs Corporation: B3500, General Electric: GE635, Univac: 1108, National Cash Register Corporation: Century 100, Apple, Inc.: MacIntosh (Motorola 68000/PowerPC and Intel X86)