Modesto Junior College
Course Outline of Record

CMPSC 216

I. OVERVIEW
The following information will appear in the 2010 - 2011 catalog

CMPSC 216 Script Programming for the Web 3 Units
Formerly listed as: CMPSC - 216: Javascript Programming for the Internet
Prerequisite: Satisfactory completion of CMPSC 204.

Developing World Wide Web applications with HTML and scripting tools such as python, javascript, ruby and perl. An introduction to creating interactive HTML documents through manipulation of the WWW DOM (Document Object Model). Designing Web-based applications, validating and processing user input, creating dynamic documents utilizing DHTML. Extensive programming projects demonstrating problem solving and implementation skills will be assigned throughout the semester. Hands-on computer assignments required.

Three maximum completions.
Field trips might be required. (A-F or P/NP - Student choice) Lecture /Lab
Transfer: (CSU, UC)

II. LEARNING CONTEXT
Given the following learning context, the student who satisfactorily completes this course should be able to achieve the goals specified in Section III, Desired Learning:

A. COURSE CONTENT

1. Required Content:

   a. Introduction to Script Programming
      i. Scripting languages Role on the Web
      ii. Creating an HTML Document
      iii. Script programming
      iv. Logic and debugging
      v. Embedding code using the < script > tag
      vi. Adding comments to documents
      vii. Creating separate source files
      viii. Incompatibility issues

   b. Variables, Functions, Objects, and Events
      i. Variable declaration and scope
      ii. Defining and calling Functions
      iii. Implementing methods through event handlers
c. Data Types and Operators
   i. Numeric data types
   ii. Boolean values
   iii. Strings
   iv. Arrays
   v. Arithmetic operators
   vi. Assignment operators
   vii. Comparison and logical operators
   viii. String operators
   ix. Operator precedence

d. Sequence, Selection and Iteration
   i. if then else statements
   ii. Switch statements
   iii. While and do while statements
   iv. For statements
   v. With statements
   vi. Continue statements

e. Windows, Frames and the Document Object Model
   i. The window object
   ii. Creating and configuring frames
   iii. Nesting frames
   iv. The location object
   v. The history object
   vi. The navigator object
   vii. Referencing frames and windows

f. Forms
   i. The common gateway interface
   ii. The tag
   iii. Form elements
   iv. The form object model
   v. Emailing form data
Dynamic HTML and Animation
i. The image object
ii. Image caching and animation
iii. Cross-browser compatibility
iv. Animation and cascading style sheets

Cookies and Security
i. State information
ii. The string object
iii. Saving state information with Query strings
iv. Saving state information with cookies
v. JavaScript Security
vi. Signed scripts and digital certificates

Debugging
i. Understanding debugging
ii. Error messages
iii. Tracing errors
iv. Locating bugs with comments
v. Language bugs and debugging resources

2. Required Lab Content:

In lab, students perform the functions listed in the lecture content section. Assignments are given that require application of the content.

a. Introduction to Script Programming
i. Scripting languages Role on the Web
ii. Creating an HTML Document
iii. Script programming
iv. Logic and debugging
v. Embedding code using the <script> tag
vi. Adding comments to documents
vii. Creating separate source files
viii. Incompatibility issues
b. Variables, Functions, Objects, and Events
   i. Variable declaration and scope
   ii. Defining and calling Functions
   iii. Implementing methods through event handlers

c. Data Types and Operators
   i. Numeric data types
   ii. Boolean values
   iii. Strings
   iv. Arrays
   v. Arithmetic operators
   vi. Assignment operators
   vii. Comparison and logical operators
   viii. String operators
   ix. Operator precedence

d. Sequence, Selection and Iteration
   i. if then else statements
   ii. Switch statements
   iii. While and do while statements
   iv. For statements
   v. With statements
   vi. Continue statements

e. Windows, Frames and the Document Object Model
   i. The window object
   ii. Creating and configuring frames
   iii. Nesting frames
   iv. The location object
   v. The history object
   vi. The navigator object
   vii. Referencing frames and windows

f. Forms
i. The common gateway interface
ii. The tag
iii. Form elements
iv. The form object model
v. Emailing form data

g. Dynamic HTML and Animation
i. The image object
ii. Image caching and animation
iii. Cross-browser compatibility
iv. Animation and cascading style sheets

h. Cookies and Security
i. State information
ii. The string object
iii. Saving state information with Query strings
iv. Saving state information with cookies
v. JavaScript Security
vi. Signed scripts and digital certificates

i. Debugging
i. Understanding debugging
ii. Error messages
iii. Tracing errors
iv. Locating bugs with comments
v. Language bugs and debugging resources

B. ENROLLMENT RESTRICTIONS

1. Prerequisites

Satisfactory completion of CMPSC 204.

2. Requisite Skills

Before entering the course, the student will be able to:

a. Identify the hardware components of a computer system such as input devices, output devices, processing devices, storage devices, communications devices and describe the functionality of each.
b. Describe the types of software used for programming development such as editors, translators/compilers, linkers, debuggers, and demonstrate operating proficiency with each type of development to

c. Describe the principles of operating systems, describe operating system interfaces, and demonstrate proficiency with using a modern operating system.

d. Describe the concepts of software engineering.

C. **HOURS AND UNITS**

<table>
<thead>
<tr>
<th>INST METHOD</th>
<th>TERM HOURS</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lect</td>
<td>36</td>
<td>2.00</td>
</tr>
<tr>
<td>Lab</td>
<td>54</td>
<td>1.00</td>
</tr>
<tr>
<td>Disc</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

D. **METHODS OF INSTRUCTION (TYPICAL)**

*Instructors of the course might conduct the course using the following method:*

1. Formal lectures by a certified instructor
2. Assigned reading and discussion of required text
3. Assigned reading of supplemental reference materials
4. Implementation of computer laboratory projects
5. The student will research network resources using technical manuals and the Internet with instructor guidance.
6. The student will demonstrate troubleshooting and problem solving techniques through hands-on projects with instructor guidance.
7. The student will analyze and present a documented solution to scenario-based case projects with instructor guidance.

E. **ASSIGNMENTS (TYPICAL)**

1. **EVIDENCE OF APPROPRIATE WORKLOAD FOR COURSE UNITS**

   *Time spent on coursework in addition to hours of instruction (lecture hours)*

   - Weekly lab assignments and projects
   - Weekly quizzes and review questions
   - Chapter Exams and projects

2. **EVIDENCE OF CRITICAL THINKING**

   *Assignments require the appropriate level of critical thinking*

   A typical project or lab:
   
   Using a scripting language the student will be able to create a web based form to accept user input. The input is then displayed and/or stored into a database.

   A typical question for PHP

   Quesiton: Explain the difference between the in_array() and array_search() functions.
   
   Answer: You can use the in_array() and array_search() functions to determine whether a value exists in an array. The in_array() function returns a Boolean value of true if a given value exists in an array. The array_search() function determines whether a given value exists in an array and returns the
index or key of the first matching element if it exists or false if it does not exist. Both functions accept two arguments: The first argument represents the value to search for, whereas the second argument represents the name of the array in which to search.

F. **TEXTS AND OTHER READINGS (TYPICAL)**


III. **DESIRED LEARNING**

A. **COURSE GOAL**

*As a result of satisfactory completion of this course, the student should be prepared to:*

create, administer, and implement a moderate sized dynamic web server using various scripting languages using the Model/View/Controller programming paradigm.

B. **STUDENT LEARNING GOALS**

*Mastery of the following learning goals will enable the student to achieve the overall course goal.*

1. **Required Learning Goals**

*Upon satisfactory completion of this course, the student will be able to:*

a. Evaluate web application requirements and develop solution strategies
b. Use graphical application planning aids such as flowcharting and object diagramming
c. Develop reusable, modular code
d. Describe the web document object model, its objects, properties and methods
e. Develop debugging techniques to troubleshoot application code
f. Evaluate application performance across web browser platforms
g. Describe the Model View Controller model, its objects, properties and methods

2. **Lab Learning Goals**

*Upon satisfactory completion of the lab portion of this course, the student will be able to:*

a. Evaluate web application requirements and develop solution strategies
b. Use graphical application planning aids such as flowcharting and object diagramming
c. Develop reusable, modular code
d. Describe the web document object model, its objects, properties and methods
e. Develop debugging techniques to troubleshoot application code
f. Evaluate application performance across web browser platforms
g. Describe the Model View Controller model, its objects, properties and methods
h. **SECOND COMPLETION:**
   i. demonstrate updated skills reflecting current industry standards as software tools, interface and functions evolve in new versions.

j. **THIRD COMPLETION:**
k. demonstrate updated skills reflecting current industry standards as software tools, interface and functions evolve in new versions.

IV. METHODS OF ASSESSMENT (TYPICAL)

A. FORMATIVE ASSESSMENT

1. Assignments
2. Exams/Quizzes
3. Projects/Labs

B. SUMMATIVE ASSESSMENT

1. Assignments
2. Exams/Quizzes
3. Projects/Labs