CURRICULUM COMMITTEE CHECKLIST

NAME OF PROGRAM: Computer Careers (COMP2145 - Web Programming) Date: 04/13/12

Step 1 Reviewed change at division meeting. [X] [ ]

Step 2 Presented as informational item at Division Chair Meeting(s) and checked if it affects other departments. Like programs must meet with Division Chairs on all affected campuses (North Mankato and Faribault).

Division Chair’s signature [Signature]

Step 3 Instructional Dean reviewed and indicated need for Curriculum Committee approval. [X] [ ]

Instructional Dean’s signature [Signature]

Step 4 Advisory Committee approval indicated in meeting minutes if necessary. Minutes provided to Curriculum Committee. [X] [ ]

Step 5 Curriculum Committee made recommendations (changes, additional approvals, etc.). If no, skip to Step 7. [ ] [ ]

Step 6 Committee’s recommendations completed. (Skip if not applicable.) [ ] [ ]

Step 7 Curriculum Committee approved. [X] [ ]

Curriculum Committee Chair’s signature [Signature]

Step 8 Minutes and necessary materials provided to VP of Academic Affairs. [ ] [ ]

Step 9 Vice President of Academic Affairs approved. [X] [ ]

Vice President of Academic Affairs’ signature [Signature]

Step 10 New Course Maximum Enrollment to Shared Governance. [ ] [ ]

Step 11 President’s approval for all changes requiring MnSCU approval. [ ] [ ]

President’s signature [Signature]
### Appendix B

**New Course or Course Change Proposal Form**

<table>
<thead>
<tr>
<th>Date of Proposal:</th>
<th>April 13, 2012</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author:</td>
<td>Peter Johnson</td>
</tr>
<tr>
<td>Proposal Type:</td>
<td>*New Course</td>
</tr>
<tr>
<td>Modify Course:</td>
<td>YES</td>
</tr>
<tr>
<td>Delete Course:</td>
<td></td>
</tr>
<tr>
<td>Contact for the Course:</td>
<td>Peter Johnson</td>
</tr>
</tbody>
</table>

**Course Designator, Number and Title** (i.e.: ACCT 1800, Business Law):
**COMP2145 – Web Programming**

Number of Credits: 4

**Prerequisites:**
- C or better in COMP1140
- C or better in COMP1130 or equivalent programming experience
- Successful completion of Spreadsheet/Database Integration (COMP1125 with a C or higher), or a working knowledge of Access, Oracle, SQL Server, or mySQL

**Course Description:**
This course covers the popular server-side language PHP and Drupal, a popular CMS (Content Management System). It includes important language concepts such as data types, control statements, debugging techniques, the use of SQL (Standard Query Language). PHP will give the student experience with LAMP (Linux, Apache, MySQL, and PHP). (Prerequisites: C or better in COMP1140, C or better in COMP1130 or equivalent programming experience, Successful completion of Spreadsheet/Database Integration (COMP1125 with a C or higher), or a working knowledge of Access, Oracle, SQL Server, or mySQL.

**Grading Method:** Grade YES Pass/Fail

**Scheduling:**
- Fall X
- Spring
- Summer
- Alternate Years
- Variable
- On Demand

**Instructional Type:**
- Lecture X
- Lab
- Lecture/Lab
- Internship
- Seminar

*Class Maximum:* (For New Courses Only) / All Unlimited faculty members of a program or discipline must sign.

<table>
<thead>
<tr>
<th>Faculty Name</th>
<th>Faculty Signature</th>
<th>Class Max</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dean's Name</th>
<th>Dean's Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>L. S. Smith</td>
<td></td>
<td>1/10/13</td>
</tr>
</tbody>
</table>

If there is not enough space provided, please use the back of this form for additional signatures or click on a row with the right button of the mouse, select insert and then select insert rows below to add rows to the table.

Is this Course Proposed as a Liberal Arts Course: Yes No X

If Yes, Which MnTC Area/Areas Will it Fulfill (http://www.mntransfer.org)?

Is This Course a Requirement/Elective for a Specific Program or Programs? Yes X No

If Yes, Which Program(s)? Information Systems (AAS & Diploma) – Elective Web Programming (Certificate) - Requirement

Describe What is Changing/Being Added, and the Rationale:
Description & Competencies – To update the course's content.

What Impact Will This New Course or Change Have on Other Programs or Areas? None

➤ Attach Common Course Outline to this Form.
Web Programming
Course Outcome Summary

Course Information
Organization South Central College
Developers Peter Johnson
Development Date 5/7/2008
Revised Date 4/16/2012
Course Number COMP2145
Potential Hours of Instruction 64
Total Credits 4

Description
This course covers the popular server-side language PHP and Drupal, a popular CMS (Content Management System). It includes important language concepts such as data types, control statements, debugging techniques, the use of SQL (Standard Query Language). PHP will give the student experience with LAMP (Linux, Apache, MySQL, and PHP).

Prerequisites: COMP1140 - Web Development with a C or higher, or a working knowledge of HTML, CSS, and FTP. COMP1130 - Programming Fundamentals with a C or higher, or a working knowledge of at least one programming language.

It is strongly recommended that you have a minimum typing speed of at least 35 wpm as well as a working knowledge of Microsoft Access (COMP1125).

Types of Instruction

<table>
<thead>
<tr>
<th>Instruction Type</th>
<th>Contact Hours</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online/lecture</td>
<td>64</td>
<td>4</td>
</tr>
</tbody>
</table>

Prerequisites
C or better in COMP1140
C or better in COMP1130 or equivalent programming experience

Exit Learning Outcomes
Core Abilities
A. Written and oral communication. Students will communicate effectively in a range of social, academic, and professional contexts using a variety of means, including written, oral, numeric/quantitative, graphic, and visual modes of communication.
B. Critical and Creative Thinking. Students will develop the disposition and skills to strategize, gather, organize, create, refine, analyze, and evaluate the credibility of relevant information and ideas.
C. Ethical reasoning and action. Students will develop ethical and social responsibility to self and others, and will collaborate with others to address ethical and social issues in a sustainable manner.

Competencies
1. Install and use PHP on a local server.
Learning Objectives
a. Describe Open Source software and why it is effective for improved software development.
b. Draw a picture describing the relationship between client/server objects used by PHP and mySQL.
c. Install PHP and mySQL and an Apache web server.
d. Write a simple test program using PHP on the local server (http://localhost/ )
e. Establish a working environment for PHP web page development.
f. Use variables, constants, and environment variables in a PHP program.
2. Utilize HTML forms and PHP to get information from the user.
   Learning Objectives
   a. Create an HTML reference page showing all the form objects.
b. Use GET and POST to send variables to a PHP web page.
c. Use isset() to check for the existence of variables.
d. Use urlencode() and urldecode() to pass strings of words via the URL (method="get")
3. Utilize decision making constructs in a PHP program.
   Learning Objectives
   a. Use the if else construct to make simple decisions.
b. Use the switch statement to make multiple comparisons on a single variable.
c. Use the for loop to iterate through a list
   d. Convert a for loop into a do while loop to demonstrate the similarities.
4. Use functions and arrays as part of your code design.
   Learning Objectives
   a. Write and use a function that receives parameters by value.
b. Store the value of a value returned by a function in a variable.
c. Demonstrate the use of indexed arrays.
d. Demonstrate the use of associative arrays.
e. Demonstrate the use of both local and global variables in a program.
f. Use the keyword static to maintain a value inside of a function.
g. Use the include( ) function to access functions stored in external files.
5. Utilize two techniques to overcome the statelessness of the Web
   Learning Objectives
   a. Utilize regular expressions to validate input from user.
b. Use regular expressions to prevent SQL injection 
c. Use hidden fields to pass information to php modules.
d. Use PHP Sessions to create persistent variables that can be used by multiple requests from a client browser session.
6. Use File I/O to read and write to files.
   Learning Objectives
   a. Use file functions to read information from disk files.
b. Use file functions to write data to disk files.
c. Get file property information from the directory.
d. Copy, rename and delete files.
7. Install mySQL and use phpMyAdmin
   Learning Objectives
   a. Connect to a database.
b. Use mysql_query() to perform SQL statements.
c. Use phpMyAdmin to administer databases, tables, and fields.

8. **Use SQL**
   
   **Learning Objectives**
   a. Create a table using SQL
   b. Add new data to a table using SQL
   c. Update records using SQL
   d. Delete records using SQL
   e. Query multiple tables using simple joins
   f. Query multiple tables using inner and outer joins
   g. Use stored procedures
   h. Demonstrate how stored procedures can prevent security breaches.

9. **Install a CMS such as Drupal**
   
   **Learning Objectives**
   a. Install Drupal on a local web server.
   b. Install Drush.
   c. Set up users with different roles and permissions.
   d. Change and modify themes.

10. **Create and Edit Content**
    
    **Learning Objectives**
    a. Describe what a node is.
    b. Utilize URL aliases.
    c. Create a 'Contact Us' page.
    d. Create a menu system.
    e. Add content nodes to a menu item.
    f. Create a new content type.

11. **Extract Data for Different Views**
    
    **Learning Objectives**
    a. Install new modules.
    b. Use shortcuts on the menu bar.
    c. Find content that doesn't have a menu item.
    d. Update URL aliases for a specific content type.
    e. Create a new SQL view.
    f. Add new fields to the SQL view.
    g. Rewrite the output of a field to include other fields.
    h. Adjust the order of content in a view.
    i. Add and configure a date field.

12. **Set up a Taxonomy**
    
    **Learning Objectives**
    a. Give a real-life example of the use of taxonomy.
    b. Add taxonomy vocabulary with terms.
    c. Add a taxonomy term reference field to a content type.
    d. Link a Views field to other related content.
    e. Add an autocomplete (free tagging) field to a content type.
13. **Utilize Blocks and Permissions**
   **Learning Objectives**
   a. Describe the difference between a block and a region.
   b. Hide user login fields and other blocks.
   c. Add a "log in" option to a user menu.
   d. Configure search.
   e. Set up a cron run.
   f. Describe how permissions relate to roles.
   g. Demonstrate different text formats.

14. **Implement a WYSIWYG Editor and Media Handler**
   **Learning Objectives**
   a. Configure HTML filters.
   b. Install a WYSIWYG editor for better UX.
   c. Describe how program libraries work
   d. Modify the WYSIWYG CSS editor settings.
   e. Install a media handling program.
   f. Coordinate the media handling program with the WYSIWYG editor.
   g. Configure styles available with the WYSIWYG editor.
   h. Troubleshoot line break/paragraph problems with the WYSIWYG
   i. Set up a custom 404 "page not found" error page.

15. **Use CSS to Theme a Site**
   **Learning Objectives**
   a. Inspect any element on a page using browser tools.
   b. Apply CSS styles to the proper selector.
   c. Describe what the CSS Injector module does.
   d. Add a custom CSS stylesheet to certain pages.
   e. Hide breadcrumbs (or any element) using CSS.
   f. Use Drush to update modules.
   g. Change the font size of menu items.

16. **Create a Module using PHP Coding**
   **Learning Objectives**
   a. Describe why the site cache has to be cleared.
   c. Implement at least three different hook functions.
   d. Display a block using hook_block_view( ).
   e. Describe the naming conventions used with hook functions using examples.
   f. List eight coding standards.
   g. Describe why coding standards are so important with an open-source project.
   h. Use the l( ) and t( ) functions appropriately.

17. **Customize and Manipulate Forms & Menus**
   **Learning Objectives**
   a. Create an embedded form.
   b. Add usability using the #states attribute.
   c. Describe what a menu callback does.
d. Add a new page without a menu item.
e. Pass arguments using the path.
f. Create dynamic titles with a title callback.
g. Modify page output using hook_page_alter() 
h. Modify menu output using hook_menu_alter() 

18. Customize and Manipulate Themes
   Learning Objectives
   a. Invoke a theme function.
b. Describe how the theme registry works.
c. Create a template file.
d. Use preprocessing functions.
e. Add CSS files in theme functions and template files.
f. Override the theme functions using hook_theme_alter()

19. Interact with the Database
   Learning Objectives
   a. Build a database table in an external application.
b. Write an INSERT SQL query using the Database API.
c. List the four benefits of using the Database API.
d. Write a DELETE SQL query using the Database API.
e. Write a dynamic SELECT query and paginate the result.
f. Add utility queries for CRUD.

20. Work with Users
    Learning Objectives
    a. Modify the user settings form.
b. Save data to an authenticated user.
c. Respond to a user login/logout.
d. Store anonymous user data in a cookie.
e. Store user data in a session variable.

21. Work with Nodes
    Learning Objectives
    a. Load and view multiple nodes at once.
b. Manipulate node data with node_save().
c. Add custom data to a node with hook_node_insert() and hook_node_update().
d. Add a custom view mode.
e. Control node-based access using hook_node_access().
f. Add a new node type with a module.
g. Add a custom field formatter.

22. Incorporate JavaScript and jQuery
    Learning Objectives
    a. Include a JavaScript file on every page of the site.
b. Use jQuery to add interactivity to a page.
c. Manipulate CSS using jQuery.
d. Use hook_library_alter() to see what JavaScript libraries are available.
e. Use Ajax to load dynamic HTML, CSS, and JavaScript.
f. List at least four JavaScript utilities that are included in the core.