1. **COURSE ID:** CIS 380  
**TITLE:** PHP Programming  
**Units:** 3.0 units  
**Hours/Semester:** 48.0-54.0 Lecture hours; and 16.0-18.0 Lab hours  
**Method of Grading:** Grade Option (Letter Grade or P/NP)  
**Recommended Preparation:**  
Eligibility for ENGL 838 or ENGL 848.  
Completion of CIS 111 or CIS 254

2. **COURSE DESIGNATION:**  
Degree Credit  
Transfer credit: CSU

3. **COURSE DESCRIPTIONS:**  
**Catalog Description:**  
Comprehensive course in PHP (hypertext preprocessor scripting language). Includes writing server-side PHP scripts for the Web, procedural and object-oriented programming, forms and browser I/O, an introduction to SQL statements and the MySQL database, and advanced topics such as creating dynamic Web content with PHP and MySQL, sessions, Web services, e-commerce, and authentication with PHP. Intended for students with previous programming experience.

4. **STUDENT LEARNING OUTCOME(S) (SLO'S):**  
Upon successful completion of this course, a student will meet the following outcomes:  
1. Develop server-side PHP scripts for publishing on the Web.  
2. Employ control structures, functions, arrays and hashes to create robust PHP applications.  
3. Design rigorous code for validation and authentication with attention to security and performance.  
4. Write SQL commands and use a MySQL database to develop data-driven Web applications  
5. Create an advanced project using PHP and a MySQL database.

5. **SPECIFIC INSTRUCTIONAL OBJECTIVES:**  
Upon successful completion of this course, a student will be able to:  
1. Develop server-side PHP scripts for publishing on the Web.  
2. Employ control structures, functions, arrays and hashes to create robust PHP applications.  
3. Design rigorous code for validation and authentication with attention to security and performance.  
4. Write SQL commands and use a MySQL database to develop data-driven Web applications.  
5. Create an advanced project using PHP and a MySQL database.

6. **COURSE CONTENT:**  
**Lecture Content:**  
1. Introduction  
   A. History of PHP  
   B. Limitations of COI Programming  
   C. Dynamic Web Pages  
   D. PHP and HTML  
   E. Web Servers and Server-Side Programming  
   F. PHP and Databases  
   G. UNIX/Linux Environment  
2. Data and Operations  
   A. Data Types (No Explicit Type Declarations)  
   B. Variables  
   C. Constants  
   D. Operators  
   E. Expressions  
   F. Operator Precedence  
3. Conditional Statements and Control Structures  
   A. IfElse  
   B. Switch  
   C. While
D. Do/While
E. For
F. Foreach
G. Break
H. Continue
I. Exit
J. Die
K. Return

4. Functions
   A. Scope
   B. Static Variables
   C. Arguments
   D. Recursion
   E. Dynamic Function Calls
   F. Exception Handling

5. Arrays and Hashes
   A. Single-Dimensional Arrays
   B. Indexing Arrays
   C. Initializing Arrays
   D. Multi-Dimensional Arrays
   E. Casting Arrays
   F. Hashes

6. Classes and Objects
   A. Defining a Class
   B. Constructors
   C. Accessing Properties and Methods
   D. Static Class Members
   E. Access Types
   F. Binding

7. Input/Output and Disk Access
   A. HTTP Connections
   B. Writing to the Browser
   C. Getting Form Input
      a. GET Method
      b. POST Method
   D. Passing Arrays in Forms
   E. Cookies
   F. File Uploads
   G. Reading/Writing to Files
   H. Sessions
      a. Browser

8. MySQL Databases
   A. Introduction to SQL Syntax
   B. Designing and Creating a Table in MySQL
   C. MySQL Data Types
   D. MySQL Metadata
   E. Inserting Values into a MySQL Table
   F. Retrieving Data from a MySQL Table
   G. Deleting Data from a MySQL Table

9. Creating Dynamic Content with PHP and MySQL
   A. Connecting to MySQL
   B. Extracting Data Using PHP
   C. Populating Form Menus Using Data from MySQL
   D. Putting Returned Records into a Table
   E. Authentication with PHP
      a. Using Sessions
      b. Using HTTP Header Authentication

10. Advanced Topics
    A. Web Services
    B. E-Commerce
Lab Content:
Write PHP applications using variables, data types, strings and methods.
Write PHP applications using loops, arrays, hashes, and sorting.
Build PHP applications using best practices.
Build PHP applications employing cookies and sessions.
Build PHP applications that use regular expressions for validation.
Write object-oriented PHP applications.
Create and manage a MySQL database.
Build dynamic data-driven websites with PHP and MySQL.
Employ TDD (Test Driven Development) and use PHP to write tests to validate an application's behavior.

7. REPRESENTATIVE METHODS OF INSTRUCTION:
Typical methods of instruction may include:
A. Lecture
B. Lab
C. Activity
D. Directed Study
E. Discussion
F. Observation and Demonstration
G. Other (Specify): Lecture will be used to introduce new topics; Teacher will model problem-solving techniques; Class will solve a problem together, each person contributing a potential "next step"; Students will participate in short in-class projects (in teacher-organized small groups) to ensure that students experiment with the new topics in realistic problem settings; Teacher will invite questions AND ANSWERS from students, generating discussion about areas of misunderstanding; Teacher will create and manage an Internet conference for discussion of course topics; and Students will work in small groups to solve programming assignments.

8. REPRESENTATIVE ASSIGNMENTS
Representative assignments in this course may include, but are not limited to the following:

Writing Assignments:
Weekly programming assignments.

Reading Assignments:
Reading assignments accompanied by self-test questions and running code examples.

Other Outside Assignments:
Studying posted lecture notes and relevant handouts. Computer programming assignments. The reading assignment frames the concepts covered and provides the basic knowledge necessary to do the self-test questions and understand the sample code. The lecture notes and handouts provide a more in-depth look at topics and distills the information down to what the faculty thinks is most important. The computer programming assignments provide hands on practice of the concepts covered in the readings.

9. REPRESENTATIVE METHODS OF EVALUATION
Representative methods of evaluation may include:
A. Exams/Tests
B. Group Projects
C. Homework
D. Lab Activities
E. Projects
F. Quizzes
G. Written examination
H. Bi-weekly quizzes (short answer--from textbook material) to provide feedback to students and teacher; Assessment of student contributions during class discussion and project time; Individual programming assignments; Midterm and Final exams (short answer - textbook material), general problem solving (similar to in-class work), short program segments (similar to programming assignments); Assessment of group participation on course projects, including peer-assessment of participation and contribution to the group effort.

10. REPRESENTATIVE TEXT(S):
Possible textbooks include:

**Origination Date:** November 2014  
**Curriculum Committee Approval Date:** January 2015  
**Effective Term:** Fall 2015  
**Course Originator:** Melissa Green