



South Central College

# COMP 2145 Web Programming

## Common Course Outline

### Course Information

**Description** This course teaches you one of the popular server-side programming languages so that you can design and build secure web applications. In this class, you will learn the principles of the client-server architecture and protocols that govern the network communication and data transfer. You will implement a web framework to create dynamic websites that use databases to store and process data. You will design and create user-facing web applications with security and user experience in mind. (Prerequisites: COMP 1130 with a C [2.0] or higher, and COMP 1140 with a C [2.0], OR higher OR instructor permission.)

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

**Total Credits** 4

**Total Hours** 64

### Types of Instruction

Instruction Type	Credits/Hours
Lecture	4/64

### Pre/Corequisites

Prerequisite C (2.0) or better in COMP1140

Prerequisite C (2.0) or better in COMP1130

OR Instructor permission

### Institutional Core Competencies

**Civic Engagement and Social Responsibility** - Students will be able to demonstrate the ability to engage in the social responsibilities expected of a community member.

**Communication** - Students will be able to demonstrate appropriate and effective interactions with others to achieve their personal, academic, and professional objectives.

**Critical and Creative Thinking** - Students will be able to demonstrate purposeful thinking with the goal of using a creative process for developing and building upon ideas and/or the goal of using a critical process for the analyzing and evaluating of ideas.

### Course Competencies

**1. Establish a working environment for server-side development.**

**Learning Objectives**

Employ programming tools necessary for server-side development.  
Use a server development environment.  
Write a simple test program using the local server.

**2. Describe the client-server architecture.**

**Learning Objectives**

Distinguish the various internet and web protocols.  
Describe the HTTP request-response cycle.  
Compare and contrast static and dynamic site characteristics.  
Examine the messaging data contained in the HTTP requests and responses.

**3. Create code in a programming language that includes primitive data types, references, variables, expressions, assignments, and control structures.**

**Learning Objectives**

Demonstrate statements and expressions that utilize variables, primitive data types, references, variables, expressions, and assignments.  
Use the if/else construct to make decisions.  
Use looping constructs to iterate through a list.

**4. Use functions and arrays as part of your code design.**

**Learning Objectives**

Write and use functions that receive parameters and return values.  
Demonstrate the use of arrays.

**5. Use file I/O to read and write to files.**

**Learning Objectives**

Use file functions to read information from disk files.  
Use file functions to write data to disk files.  
Get file property information from the directory.  
Copy, rename, and delete files.

**6. Apply web security best practices.**

**Learning Objectives**

Explain browser and web security model concepts, including same-origin policy, web sessions, and secure communication channels.  
Implement input validation.  
Use methods to prevent SQL injection.  
Employ access control based on user authentication and authorization.  
Implement tools to assess the potential vulnerabilities of your application.  
Apply sanitization to all incoming data.  
List resources and publications that help you keep track of the security threats in the web programming domain.

**7. Utilize techniques to overcome the statelessness of the Web.**

**Learning Objectives**

Create a simple program that uses persistence to save data across multiple executions.  
Store anonymous user data in a cookie.

**8. Install a database and use database interface tools.**

**Learning Objectives**

Set up a database connection.  
Create a database schema fulfilling the needs of the application.  
Use database interface tools to administer databases, tables, and fields.

**9. Apply database programming to store and retrieve data.**

**Learning Objectives**

Create tables.  
Query multiple tables using simple joins.  
Query multiple tables using inner and outer joins.  
Use stored procedures.  
Demonstrate how stored procedures can prevent security breaches.

**10. Implement CRUD (Create, Read, Update, Delete) operations.**

**Learning Objectives**

Apply database CRUD (Create, Read, Update, Delete) operations from the web application.  
Utilize GET and POST methods in your programs.

**11. Create dynamic websites.**

**Learning Objectives**

Create websites that respond differently based on user interaction.  
Develop applications with custom user experience in mind.  
Store and retrieve data for an authenticated user.

**12. Create web forms.**

**Learning Objectives**

Identify best practices for designing web forms.  
Apply server-side programming principles to build a usable form, including input validation, security, user experience (UX), and database programming.

**13. Implement a web framework for server-side programming.**

**Learning Objectives**

Compare and contrast web frameworks  
Employ a framework for common tasks to ease the development process.  
Use a framework for sessions, users, and authentication.  
Utilize a framework for security reasons to prevent threats including Cross-Site Scripting (XSS), SQL injection, Cross-Site Request Forgery (CSRF).  
Implement routing.  
Simplify database access with Object-Relational Mapper (ORM).  
Use templating libraries for data access and presentation.