

6. **Student Learning Outcomes** (Identify 1-6 expected learner outcomes using active verbs.)

Upon successful completion of the course, the student will be able to:

1. Develop server-side Ruby scripts for publishing on the Web;
2. Employ control structures, methods, procs, arrays and hashes to create Ruby programs;
3. Explain object-oriented programming and input/output processing and apply these concepts to develop dynamic interactive Ruby applications;
4. Discuss Model-View-Controller architecture and its relationship to Ruby on Rails applications.
5. Use SQL commands and the MySQL database together with Ruby;
6. Create an advanced project using MySQL, Ruby and the Ruby on Rails framework.

7. **Course Objectives** (Identify specific teaching objectives detailing course content and activities. *For some courses, the course objectives will be the same as the student learning outcomes. If this is the case, please simply indicate this in this section).*

See Student Learning Outcomes

8. **Course Content** (Brief but complete topical outline of the course that includes major subject areas [1-2 pages]. Should reflect all course objectives listed above. In addition, you may attach a sample course syllabus with a timeline.)

See attached topical outline

9. **Representative Instructional Methods** (Describe instructor-initiated teaching strategies that will assist students in meeting course objectives. Include examples of out-of-class assignments, required reading and writing assignments, and methods for teaching critical thinking skills.)

The course will include the following instructional methods as determined appropriate by the instructor:

- 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.

10. **Representative Methods of Evaluation** (Describe measurement of student progress toward course objectives. Courses with required writing component and/or problem-solving emphasis must reflect critical thinking component. If skills class, then applied skills.)

- Bi-weekly quizzes (short answer--from textbook material) to provide feedback to students and teacher on objectives 1 - 5;
- Assessment of student contributions during class discussion and project time to assess objectives 1-5;
- Individual programming assignments to assess objectives 1, 2, 3, and 6;
- Midterm and Final exams (short answer (textbook material), general problem solving (similar to in-class work), short program segments (similar to programming assignments)) to assess objectives 1-5;

- Assessment of group participation on course projects, including peer-assessment of participation and contribution to the group effort to assess objective 6.

11. **Representative Text Materials** (With few exceptions, texts need to be current. Include publication dates.)

Programming Ruby, Thomas, Pragmatic Bookshelf ISBN 0-9745140-5-5, 2006

The Ruby Programming Language, Flanagan & Matsumoto, O'Reilly ISBN 0-5965161-7-7, 2008

Ruby By Example, Concepts and Code, Baird, No Starch Press ISBN 1-5932714-8-4, 2007

Agile Web Development with Rails, Thomas & Hansson, Pragmatic Bookshelf ISBN 0-9776166-3-0, 2006

Beginning Ruby: From Novice to Professional, Cooper, Apress ISBN 1-5905976-6-4, 2007

Ruby for Rails: Ruby Techniques for Rails Developers, Black, Manning Publications ISBN 1-9323946-9-9, 2006

Prepared by:

(Signature)

Email address:

greenm@smccd.edu

Submission Date:

CIS 113 – Internet Programming: Ruby Topical Outline

For Objective 1

1. Introduction
 - a. History of Ruby
 - b. Dynamic Web Applications
 - c. Ruby Installation
 - d. Web Servers and Server-Side Programming
 - e. Ruby and Databases
 - f. UNIX/Linux Environment

For Objectives 2 - 5

2. Data and Operations
 - a. Data Types
 - b. Variables
 - c. Constants
 - d. Operators
 - e. Expressions
 - f. Operator Precedence
 - g. Scope

3. Conditional Statements and Control Structures

- a. If Else
- b. Case
- c. While
- d. Do
- e. Foreach
- f. For In
- g. Redo

4. Methods

- a. Method Definition
- b. Methods and Blocks
- c. Calling a Method
- d. Method Parameters
- e. Method Return Values
- f. Exception Handling

5. Arrays and Hashes

- a. Indexing Arrays
- b. Initializing Arrays
- c. Array Class
- d. Hashes
- e. Blocks and Iterators

6. Classes and Objects

- a. Defining a Class
- b. Objects and Attributes
- c. Access Control
- d. Inheritance and Messages
- e. Class Variables and Class Methods
- f. Modules and Namespaces
- g. Mixins
- h. Method Objects

7. Web Interaction and File Access

- a. Ruby HTML Code Generation
- b. CGI Code Generation
- c. Templating Systems
- d. Form Input
- e. Validation and Regular Expressions
- f. Cookies
- g. Sessions
- h. Reading/Writing Files

For Objectives 4, 5

8. SQL and the MySQL Database

- a. Introduction to SQL Syntax
- b. Designing and Creating a Table in MySQL
- c. MySQL Data Types

9. Dynamic Applications with Ruby on Rails and MySQL

- a. Model-View-Controller Architecture

- b. Connecting to MySQL
- c. Active Record
 - 1. Object-Relational Mapping
- d. Action Controller
- e. Action View
- f. Authentication
- g. Testing and Debugging