Salem STATE | UNIVERSITY

Computer Science Department
cs.salemstate.edu

CSC 278 Scripting Techniques **4 cr.**

Instructor: TBA **Office:** location **Phone:** (978) 542-extension
email: TBA@salemstate.edu **Office Hours:** days and times

Section	Time	Room	Final Exam
nn	days and times	location	date and time
Lnn	days and times	location	

Catalog description:

This course presents rapid application development (RAD) techniques and their implementation using modern scripting languages. Methods for defining problems and their solutions will be examined, including task analysis and the development of design criteria. The course investigates the design of modern scripting languages, emphasizing the use of their particular attributes for developing solutions to complex problems. Fundamental programming language paradigms, type systems, and memory allocation and management strategies are presented and discussed, followed by comparative analysis of the languages utilized in this course and its prerequisite. Three hours of lecture and three hours of scheduled laboratory per week, plus programming work outside of class.

Prerequisites: CSC 115 or CSC 202J.

Course Goals:

The purpose of this course is for students to:

CG01: develop a greater understanding of the software development process;
CG02: understand the role of scripting languages in software development;
CG03: present the use of scripting languages in developing web applications;
CG04: examine the use of scripting languages for the rapid development of scientific applications;
CG05: implement one or more large-scale projects using a scripting language.

Course Objectives:

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

CO01: determine if a scripting language is appropriate for a particular problem and if so, to select a specific language that is well suited to the particular problem;
CO02: select appropriate software engineering methods and RAD techniques for developing applications using a scripting language;
CO03: select appropriate algorithms, data structures, and language features for the solution of a complex problem;
CO04: understand and employ common types of code reuse;
CO05: understand and utilize open-source modules for complex problem solving in computational science or artificial intelligence;
CO06: produce clear documentation for problems assigned in the course and their solutions.

Program Outcome vs. Course Objectives matrix

Program Objective	**CO01**	**CO02**	**CO03**	**CO04**	**CO05**	**CO06**
PO-A: apply knowledge of computing and math	✓	✓	✓	✓	✓	✓
PO-B: analyze a problem and define its computing requirements	✓	✓	✓			✓
PO-C: design, implement and evaluate applications		✓	✓	✓	✓	
PO-D: function effectively in teams to accomplish a common goal						✓

Program Objective	**CO01**	**CO02**	**CO03**	**CO04**	**CO05**	**CO06**
PO-E: professional, ethical, and social responsibilities						
PO-F: communicate effectively with a range of audiences						
PO-G: local and global impact of computing on people and society						
PO-H: need for continuing professional development						
PO-I: use current techniques, skills, and tools	✓	✓	✓	✓	✓	✓
PO-J: apply theory and principles to model and design systems			✓			
PO-K: apply design and development principles in constructing software				✓		
note - full statements of the **P**rogram **O**utcomes (objectives) for the Computer Science Major can be found in the document *Computer Science Major Program Educational Objectives and Program Outcomes* on the Assessment page of the Computer Science Major (cs.salemstate.edu)						

Course Topics:

- Programming language concepts
  - programming paradigms **(4)**
    - procedural
    - object-oriented
    - functional
    - declarative
    - multi-paradigm
    - emerging and/or specialized paradigms
  - type systems **(3)**
    - "strongly-typed" vs "weakly-typed" vs. "type safe"
    - static (compile-time) type checking
    - dynamic (run-time) type checking
  - memory allocation and management **(1)**
    - static vs. dynamic
    - direct vs. indirect
- Software engineering environment and methodologies **SE1(1), SE4(1), SE5(2), PL1(2)**
  - The development of task requirements and software specifications.
  - Introduction to effective methods for software development and prototyping.
  - Tools for project planning and revision control
  - Review programming language development, with an emphasis on the need for scripting languages. Compare and contrast high level languages, such as C++ and Java, with scripting languages, such as JavaScript and Python.

- Developing Web-Base Applications Using Scripting Languages **HC1(1), HC2(2), HC5(1), HC6(1), NC4(1), NC5(4), SE1(1), SE3(1), PF5(2)**
  - The Document Object Model (DOM)
  - DOM Scripting
  - Frames, forms, cookies, and alarms
  - Processing events
  - Dynamic HTML

- Developing Scientific Applications Using Scripting Languages
  - Tools for application development: IDE's, interpreters, compilers, and applet builders **SE3 (1)**
  - Examination of a popular language currently used for scientific applications **PL5(1), PL6(1), PL7(1)**
  - Cross-platform GUI Development **HC5(1), HC6(2)**
  - Multi-threading using scripting languages **OS3(1)**
  - According to student and instructor interest, examine an application of scripting techniques in one or more of the following areas:
    - Artificial intelligence **IS2(2)**
    - Image processing and computer vision **GV11(2)**

- Algorithms for data collection from the world-wide web **IS8(2)**
- Scientific computing using existing libraries **CN1(2) or CN3(2)**

Assignments: Students will submit one short paper on the software development process, complete six to eight laboratory exercises to learn different scripting languages, and test different software development methods. Homework, in the range of four to six assignments, will be given to reinforce learning of concepts and language features.

Quizzes, Tests and Examinations: A midterm examination plus a comprehensive final examination that will be administered during the final exam period.

Grading: The final grades will be determined according to the formula: final 20%, midterm 20%, laboratory work 40%, homework 20%.

Course Objective / Assessment Mechanism matrix

	Homework Assignments	Midterm Examination	Labs	Final Examination
CO01	✓	✓	✓	✓
CO02	✓	✓	✓	✓
CO03	✓	✓	✓	✓
CO04	✓	✓	✓	✓
CO05	✓		✓	
CO06	✓		✓	

Bibliography:

Arnowitcz, Jonathan; Arent, Michael; Berger, Nevin. **Effective Prototyping for Software Makers**, Kaufmann, 2006.
Beazley, David M. **Python Essential Reference. 4th ed..** Sams, 2009.
Downey, Allen . **Learning Perl the Hard Way**, Green Tea Press, 2010.
Flanagan, David **JavaScript: The Definative Guide**, 6th ed., O'Reilly, 2011.
Langtangen, Hans Petter, **Python Scripting for Computational Science**. **Second Edition.** Springer; 2010)
Lutz, Mark, Ascher, David. **Learning Python. Third Edition.** O'Reilly, 2007.
Keith, Jeremy. **DOM Scripting: Web Design with JavaScript and the Document Object Model.** friends of ED, 2005.

Pilgrim, Mark **Dive Into Python 3**, Create Space, 2010.
Rappin, Noel; Dunn, Robin. **WxPython in Action**. Manning Publications, 2006.
Stefanov, Stoyan **Object-Oriented JavaScript: Create Scalable, reusable high-quality JavaScript applications and libraries**, Packt, 2008.
Wyke-Smith, Charles **Scriptin' with JavaScript and Ajax**,New Riders, New Riders, 2010.

Academic Integrity Statement:

"Salem State University assumes that all students come to the University with serious educational intent and expects them to be mature, responsible individuals who will exhibit high standards of honesty and personal conduct in their academic life. All forms of academic dishonesty are considered to be serious offences against the University community. The University will apply sanctions when student conduct interferes with the University primary responsibility of ensuring its educational objectives." Consult the University catalog for further details on Academic Integrity Regulations and, in particular, the University definition of academic dishonesty.

The Academic Integrity Policy and Regulations can be found in the University Catalog and on the University website (http://catalog.salemstate.edu/content.php?catoid=13&navoid=1295#Academic_Integrity). The formal regulations are extensive and detailed - familiarize yourself with them if you have not previously done so. A concise summary of and direct quote from the regulations: "Materials (written or otherwise) submitted to fulfill academic requirements must represent a student's own efforts". *Submission of other's work as one's own without proper attribution is in direct violation of the University's Policy* and will be dealt with according to the University's formal Procedures. *Copying without attribution is considered cheating in an academic environment - simply put,* ***do not do it!***

University-Declared Critical Emergency Statement:

In the event of a university-declared emergency, Salem State University reserves the right to alter this course plan. Students should refer to www.salemstate.edu for further information and updates. The course attendance policy stays in effect until there is a university-declared critical emergency.

In the event of an emergency, please refer to the alternative educational plans for this course, which will be distributed via standing class communication protocols. Students should review the plans and act accordingly. Any required material that may be necessary will have been previously distributed to students electronically or will be made available as needed via email and/or Internet access.

Equal Access Statement:

"Salem State University is committed to providing equal access to the educational experience for all students in compliance with Section 504 of The Rehabilitation Act and The Americans with Disabilities Act and to providing all reasonable academic accommodations, aids and adjustments. **Any student who has a documented disability requiring an accommodation, aid or adjustment should speak with the instructor immediately.** Students with Disabilities who have not previously done so should provide documentation to and schedule an appointment with the Office for Students with Disabilities and obtain appropriate services."

Note: This syllabus represents the intended structure of the course for the semester. If changes are necessary, students will be notified in writing and via email.