University of Maryland University College

School of Undergraduate Studies

Faculty Teaching Guide

CMIS 325
UNIX with Shell Programming

About CMIS 325

Listed below are important concepts and highlights about the nature of this course and its students.

- Students will have access to a UMUC UNIX account to test and deploy their projects and exercises. Instructors also have access to these accounts.
- The focus on this course is on basic UNIX commands and shell programming.
- Students should understand the concept of automating tasks in the UNIX environment.

Using this Guide

This guide is designed to help faculty members teach the course effectively as well as to help the department maintain the appropriate alignment with the program curriculum. The guide and the course syllabus should be provided to faculty members teaching the course and should be regularly reviewed and updated. When you have suggestions and recommendations for the guide, see the corresponding 999 conference discussion.

How can you use this guide to help you teach your course? It's a long-term resource to consult as you progress, but here are some immediate ideas to:

- **Plan the flow of your class—**Look at the assignments and activities in the guide—they are extended descriptions of what you will find in the model syllabus, and will help you think about timing.

- **Introduce and explain your course to students—**Deliberately point to the intended course outcomes—you'll find them right after the course description. Also be sure to explain the level and place of the course and the kind of activities the students will experience. Students need to understand the focus and purpose of their learning.

- **Choose topics for class discussions—**Besides the outcomes, look at the Learning Activities section, which may give you ideas for discussions or techniques. The Required Concepts, Skills, and Issues to Be Covered section will also remind you of content to present to achieve the outcomes.

- **Explain class assignments—**Sample language is provided for major assignments, as well as assistance with grading weights and criteria. Feel free to cut and paste this language into your explanations. The assignments are designed to fit coherently with the course design. Make sure you specify the course outcomes for each assignment. This will show students there is purpose in the assignment.

- **Monitor student progress—**The course design and guide are planned for a sequence to achieve the learning outcomes. Look at how assignments are split up or staged in the syllabus and the suggestions in the guide for evaluating student work. You may want to institute subprojects, such as deadlines for drafts or pieces of larger projects, to let you know if students are ready for the big assignments.

- **Make sure you are teaching at the right level—**Besides the explanation of the place and level of the course at the beginning of the guide, review the Bloom's taxonomy chart at the end, which gives a quick picture of the cognitive level for this course. It can help you make sure you aren't teaching an introductory course at a level beyond your students' abilities or, conversely, teaching an advanced course at too low a level.

Course Number and Title

CMIS 325: *UNIX with Shell Programming*

Course Description

Prerequisite: CMIS 141, CMIS 115, or CMIS 125. A hands-on, project-based introduction to the UNIX operating system. The aim is to use basic UNIX commands to design, create, and execute shell programs. Topics include file structures, editors, pattern-matching facilities, shell commands, and shell scripts.

Intended Course Outcomes

List the learning outcomes for this course (phrased as "After completing this course, students should be able to ...").

1. select and adapt IT technologies based on relevant and objective criteria
2. apply UNIX commands to navigate and manage file systems and data
3. create and run well-documented UNIX shell programs to simplify and automate frequent tasks

Position of Course in Curriculum

- **Role in Program and Degree Requirements**

How does this course relate to requirements (e.g., required course, one of several choices for a specific requirement, supplemental course for major, elective)?

CMIS 325 is an elective 300-level course for CMIS students. Students may choose to take this course or one of the following courses: computer architecture (CMIS 310), C++ programming (CMIS 315), or a Windows Phone 7 course (CMIS 255).

- **Program Outcomes to Which this Course Maps**

To which program-level outcomes does this course contribute? How does it contribute?

P2. Design, develop, implement, secure, and maintain software applications that meet user requirements, using current best practices and tools for all application interfaces and domains.
P4. Plan, manage, and provide appropriate documentation and communication through all phases of the software development life cycle to ensure successful implementation of an information technology (IT) project that is on time and within budget.
P6. Work independently or as an effective member of an application development team to determine and implement systems that meet customer requirements.

Students will develop powerful and useful scripts based on user requirements in accordance with best practices and current trends.

- **Relation to Hallmark Competencies, if Appropriate**

Hallmarks are the core competencies that UMUC assures each student will accomplish by the end of their degree. The hallmarks are introduced, reinforced, and emphasized in general education courses as well as the program.

Which hallmarks are addressed in this course (effective writing, information literacy, technology fluency, critical thinking, ethics, quantitative literacy, scientific literacy, historical and cultural perspectives)?

H1. Historical and Cultural Perspectives

H2. Written Communications

H3. Information Literacy

H5. Critical Thinking

H6. Technology Fluency

- **Level of Course (100, 200, 300, or 400)**

Why is the stated level appropriate for this course?

This is a 300-level course that requires previous programming skills acquired in a lower-level course and helps students build more advanced scripts designed for efficiency and batch processes. Higher-order critical thinking is required to be successful in this course.

- **Relation to or Sequencing with Other Courses**

How does this course relate to other courses (e.g., to prerequisites; as a prerequisite; where the course will fall in a recommended sequence; role as part of a group of courses)?

An introductory programming course is required before registering for this course.

- **Importance of the Course to Other Majors and/or Disciplines**

How does this course contribute to programs outside this major (e.g., requirement, related requirement or recommended elective for another major, fulfills general education requirement)?

Students who take this course may take additional Linux or UNIX administration courses available in the CMIT program. Students in other majors may take this course provided they have an appropriate prerequisite.

Approach to Course

- **Required Concepts, Skills, and Issues to Be Covered**

List the concepts, skills, and issues that must be included as central to the course (versus optional material that may be included depending on time and instructor/student interest).

As part of the redesign process, the following concepts, skills, and issues were identified as important to student success. Faculty members should design the overall course, reading assignments, weekly discussions, activities, and projects to address these in an appropriate manner to ensure that students are prepared to accomplish the graded activities. These have been arranged in a suggested weekly sequence; however, faculty should align with assigned readings and in a way that best supports the assignments and course activities.

Date	Assignments	Due Date
Week 1	**Concepts** • connecting to a UNIX environment • piping • basic UNIX commands • text editing vi part 1 • redirection • e-mail • control C • using documentation effectively **Skills** • using pipes • using basic UNIX commands • using text editors **Issues** • logging into system	
Week 2	**Concepts** • vi part 2 • files and directories • setting permissions • print • environment variables • comparative operating systems **Skills** • working with files and directories • setting permissions **Issues** • text vs. graphical view • setting permissions	
Week 3	**Concepts** • text processing commands • regular expression syntax • job control (kill) • GUI • FTP • secure FTP (DOS) **Issues** • accidents	

Week 4	**Concepts** • shell script programming • command line arguments • variable substitution • command substitution • shell script I/O • filename generation **Skills** • debugging shell programs **Issues** • debugging shell programs	
Week 5	**Concepts** • positional parameters (functions) • command substitution • quoting • here doc • selection statements • case construct **Skills** • using case construct • using "if-then-else" statements	

Week 6	**Concepts**
	• loops
	  ○ while
	  ○ for
	• math operators
	• shell variables
	
	Skills
	• writing documentation	

Week 7	**Concepts** • sed • data validations • handling bad input • history of UNIX • versions of UNIX	
Week 8	**Concepts** • cron jobs • arrays (mention only) • UNIX configurations	

Assessment and Learning Activities

- **Assessments, Projects, and Assignments**

List and describe the planned projects/assignments for this course that will fulfill the course outcomes and allow for appropriate assessment. Mark mandatory items with an asterisk (*). If an item is included in the program plan for assessment of a program-level outcome or a hallmark outcome, mark and specify the assessment activity to be required for that purpose as well (e.g., common final exam).

- conferences *
- weekly assignments *
- final project *

Faculty members are free to slightly modify the assignments' contributions to the final grade by plus or minus 5 percent. For example, weekly conferences are listed as 20 percent in the syllabus; however, faculty members are free to modify this level, provided they stay in the range between 15 percent and 25 percent.

Project Descriptions

Weekly assignments (seven)

> Your instructor will provide you weekly assignments related to the readings and discussions.

Final project

- Create and manage a phone book text data file.
- Create five correctly working programs:
    - menu
    - add
    - display
    - change
    - remove
- Each of these programs should work together as a unique application.
- Include documentation as header comment blocks for each program.

Your final project will be graded based on:

- functionality (50 percent)
- proper style: indentation, use of white space, variable names (10 percent)
- adherence to requirements (30 percent)
- user-friendly application (10 percent)

- **Learning Activities**

List and describe the learning activities for this course that will address learning outcomes and fulfill the principles of the SUS learning model (e.g., working in study groups or learning collaboratively in small groups; investigating and discussing a resource; responding to material or taking a quiz in an online course module; reviewing and commenting on another student's draft; revising a draft based on specific criteria). These may or may not be graded activities. Specify required learning activities as well as suggested default learning activities that can be used at the instructor's option.

Weekly conferences should be related to the concepts and allow students to comment on one another's work and contributions.

- **Cognitive Level and Assessing Course Outcomes**

Use the table on the following page to indicate the level at which each course outcome will be addressed and assessed in the course as per Bloom's taxonomy. (See more on Bloom's taxonomy below the table.) Mark the appropriate cell with an X to indicate the level for the outcome.

Course Outcomes Mapped to Bloom's Taxonomy

Bloom's taxonomy is a tool for classifying the cognitive demand level of instructional activities or questions. As one moves through the hierarchy from knowledge to evaluation, the activities and questions require increasingly higher-level thinking skills. See the list below this table for a summary of Bloom's taxonomy and examples of verbs corresponding to each cognitive level.

	Cognitive Demand Level – Based on Bloom's Taxonomy					
Course Outcome	**Knowledge (Lowest)**	**Comprehension**	**Application**	**Analysis**	**Synthesis**	**Evaluation (Highest)**
1. select and adapt IT technologies based on relevant and objective criteria				X		
2. apply UNIX commands to navigate and manage file systems and data			X			
3. create and run well-documented UNIX shell programs to simplify and automate frequent tasks				X		

Bloom's Taxonomy Terms

Source: Wyatt, A. T. (2001). *Bloom's taxonomy*. Retrieved September 11, 2006, from http://cs1.mcm.edu/~awyatt/csc3315/bloom.htm

Level	Type of Activity or Question	Verbs Used for Outcomes
Lowest level	**Knowledge**	define, memorize, repeat, match, record, list, recall, name, relate, collect, label, specify, cite, enumerate, recite, tell, recount
	Comprehension	restate, summarize, differentiate, discuss, describe, recognize, explain, express, identify, locate, report, retell, review, translate, paraphrase
	Application	exhibit, solve, manipulate, interview, simulate, apply, employ, use, demonstrate, dramatize, practice, illustrate, operate, calculate, show, experiment
Higher levels	**Analysis**	interpret, classify, analyze, arrange, differentiate, group, compare, organize, contrast, examine, scrutinize, survey, categorize, dissect, probe, create an inventory, investigate, question, discover, inquire, distinguish, detect, diagram, chart, inspect
	Synthesis	compose, set up, plan, prepare, propose, imagine, produce, hypothesize, invent, incorporate, develop, generalize, design, originate, formulate, predict, arrange, assemble, construct, create
	Evaluation	judge, assess, decide, measure, appraise, estimate, evaluate, rate, deduce, compare, score, value, predict, revise, choose, conclude, recommend, determine, criticize, test

Appendix A

CMIS 325: *UNIX with Shell Programming*

Course Syllabus

Faculty Contact Information

Course Materials

Rosen, K. *UNIX: The Complete Reference Unix*, 2nd ed., McGraw Hill. ISBN: 978-007-226336-7

Recommended: Robbins, A. *UNIX in a Nutshell*, 4th ed., O'Reilly. ISBN: 978-0-596-10029-2

Course Description

Prerequisite: CMIS 141, CMIS 115, or CMIS 125. A hands-on, project-based introduction to the UNIX operating system. The aim is to use basic UNIX commands to design, create, and execute shell programs. Topics include file structures, editors, pattern-matching facilities, shell commands, and shell scripts.

Course Outcomes

After completing this course, you should be able to

1. select and adapt IT technologies based on relevant and objective criteria
2. apply UNIX commands to navigate and manage file systems and data
3. create and run well-documented UNIX shell programs to simplify and automate frequent tasks

Course Introduction

This course is a hands-on, project-based introduction to the UNIX operating system. By the completion of the course, you will be able to use basic UNIX commands to design, create, and execute shell programs, and effectively work in a UNIX production environment.

You will explore the history and versions of UNIX, and how they affect UNIX shell programming. You will practice and apply commands related to manipulating file structures, pattern-matching facilities, and shell commands and editors, using production UNIX servers. In your final project, you will analyze, create, and evaluate a solution to a particular programming problem.

Grading Information and Criteria

For online classes, grades will be based on the completion of the following assignments:

Conferences	15%
Weekly assignments	60%
Final project	25%
Total	**100%**

The grading scale, based on 100 points, is as follows:

A =	90-100
B =	80–89
C =	70–79
D =	60–69
F =	0–59

Participation

By registering for a Web-based course, you have made a commitment to participate in course conferences as well as other online activities. Plan to participate regularly. Participation for this course is defined as proactive discussion in weekly conferences and discussion questions. This requires you to actively reflect on weekly readings and to develop original ideas in your responses. You are expected to demonstrate critical thinking and your understanding of the content in the assigned readings as they relate to the issues identified in the conference discussion.

You are expected to respond to a main topic each week and read other student posts. You are encouraged to respond to other students as well as to your instructor. Note that your online conference participation counts significantly toward your final grade.

When communicating with others in this class, always work to be respectful.

Final Examination

There is no final examination for this course.

Other Information

Project Descriptions

Weekly assignments (seven)

Your instructor will provide you weekly assignments related to the readings and discussions.

Final project

- Create and manage a phone book text data file.
- Create five correctly working programs:
    - menu
    - add
    - display
    - change
    - remove
- Each of these programs should work together as a unique application.
- Include documentation as header comment blocks for each program.

Your final project will be graded based on:

- functionality (50 percent)
- proper style: indentation, use of white space, variable names (10 percent)
- adherence to requirements (30 percent)
- user-friendly application (10 percent)

Academic Policies

Academic Integrity

UMUC is an academic community that honors integrity and respect for others, and it is expected that, as a member of this community, you will maintain a high level of personal integrity in your academic work at all times.

Academic dishonesty is the failure to maintain academic integrity, and includes the intentional or unintentional presentation of another person's ideas or products as your own (plagiarism); the use or the attempt to make use of unauthorized materials, information, or study aids in any academic exercise; and the performance of work for another student (cheating). All academic work you submit during your time at UMUC must be original and must not be reused in other courses.

Turnitin.com

The university has a license agreement with Turnitin, an educational tool that helps identify and prevent plagiarism from Internet resources. I may use the service in class, either by requiring you to submit assignments electronically to Turnitin, by submitting assignments on your behalf, or by providing the option for you to check your own work for originality. The Turnitin Originality Report will indicate the amount of original text in your work and whether all the material that you quoted, paraphrased, summarized, or used from another source is appropriately referenced.

If you or I submit all or part of your assignment to the Turnitin service, Turnitin will by default store that assignment in its database. The assignment will be checked for any matches between your work and other material stored in Turnitin's database. If you object to the long-term storage of your work in the Turnitin database, you must let me know no later than two weeks after the start of this class.

You have three options regarding the storage of your assignment in the Turnitin database: 1) You can do nothing; your assignment will then be stored in the Turnitin database for the duration of UMUC's contract with Turnitin; 2) You can ask me to have Turnitin store your assignment only for the duration of the semester or term, then have your assignment deleted from the Turnitin database once the class is over; or 3) You can ask me to change the Turnitin settings so that your assignment is not stored in the Turnitin database at all.

Please note: I may use other services in addition to or in place of Turnitin to check your work for plagiarism.

Course Expectations

For an eight-week course, you should expect to spend about six hours per week participating in class discussions and activities (online or onsite) and two to three times that number of hours outside class in study, assigned reading, and preparation of assignments. Courses offered in shorter formats will require more time per week. You are expected to meet the same learning outcomes and perform the same amount of work in an online course as in an onsite course. Active participation is required in all online courses, and you should expect to log in to your online course several times a week.

The following links to important academic policies and other information are provided to help you as you complete your coursework at UMUC.

Policies and Procedures

- Policy and Procedures on Affirmative Action, Equal Opportunity, and Sexual Harassment—Nondiscrimination: It is the policy of UMUC that no student or employee of the university or contractor/vendor conducting business with the university may discriminate on the basis of race, religion, color, creed, sex (including sexual harassment), marital status, age, national origin, political affiliation, mental or physical disability, or sexual orientation. Individuals who believe they have been discriminated against because of any factor protected under this policy may file a complaint of discrimination.
- Information on Support for Disabled Students
- University System of Maryland Board of Regents' Policy on Academic Integrity
- UMUC's Policy on Academic Dishonesty and Plagiarism
- UMUC's Policy on the Grade of Incomplete, Grade Pending, and Withdrawal
- UMUC's Policy on the Code of Student Conduct
- UMUC's Policy and Procedures for Review of Alleged Arbitrary and Capricious Grading

For more information about student services and more general information, visit UMUC's website at http://www.umuc.edu.

Faculty Bio

Eight-Week Course Schedule

Date	Assignments	Due Date
Week 1	**Getting Started** **Read:** • Modules 1, 2 • Rosen, chapters 1, 2, 5 (Editing with vi pp. 133-142) **Do:** • Actively participate in conferences • Assignment 1	
Week 2	**The UNIX File System** **Read:** • Module 3 • Rosen, chapter 3 **Do:** • Actively participate in conferences • Assignment 2	
Week 3	**The Command Shell** **Read:** • Module 3 • Rosen, chapter 4 **Do:** • Actively participate in conferences • Assignment 3	
Week 4	**UNIX Tools** **Read:** • Review module 4 • Rosen, chapter 19 **Do:**	

	• Actively participate in conferences • Assignment 4	
Week 5	**Unix Tools, continued** **Read:** • Module 4 • Rosen, chapter 19 **Do:** • Actively participate in conferences • Assignment 5	
Week 6	**The Shell Scripting** **Read:** • Review module 5 • Review Rosen, chapter 20 **Do:** • Actively participate in conferences • Assignment 6	
Week 7	**The Shell Scripting, continued** **Read:** • Review module 5 • Review Rosen, chapter 20 **Do:** • Actively participate in conferences • Assignment 7	
Week 8	**Process and Scheduling** **Read:** • Rosen, chapter 11 **Do:** • Actively participate in conferences and discussions • Final project due	

Course Outcomes

1.	select and adapt IT technologies based on relevant and objective criteria
2.	apply UNIX commands to navigate and manage file systems and data
3.	create and run well-documented UNIX shell programs to simplify and automate frequent tasks