Online/Blended Course Syllabus

Course Information

Course Number/Section	CS 3377.0W1, SE 3377.0W1
Course Title	C/C++ Programming in a UNIX Environment
Term	Fall 2018

Professor Contact Information

Professor	Richard Min, Ph.D. MBA, MS in CE, STM, M.Div.
Office Phone	972-883-4522
Other Phone	
Email Address	rkm010300@utdallas.edu
Office Location	ECSS 4.609
Online Office Hours	MW 1-2:30 & MW 3:45-5:30 pm, TR 11:15am-1pm (via email and/or by appointment)
Other Information	

Course Pre-requisites, Co-requisites, and/or Other Restrictions

Prerequisite: (CE 2336 or CS 2336 or TE 2336) with a grade of C or better or equivalent. (Same as SE 3377) (3-0) S

Course Description

CS 3377 C/C++ Programming in a UNIX Environment (3 semester hours) Advanced programming techniques utilizing procedural and object oriented programming in a UNIX environment. Topics include file input and output, implementation of strings, stacks, queues, lists, and trees, and dynamic memory allocation/management. Design and implementation of a comprehensive programming project is required.

Student Learning Objectives/Outcomes

1. Ability to use the UNIX operating system interactively as a user (commands)
2. Ability to express algorithmic solutions using shell scripting (utilities)
3. Ability to understand and use regular expressions
4. Ability to use the UNIX programming environment (editor, compiler and linker)
5. Ability to understand UNIX processes (creation and control)
6. Ability to perform input/output of binary files
7. Ability to use interprocess communication (pipes, sockets and signals)
8. Ability to understand the UNIX file system
9. Ability to understand and use version control system

Required Textbooks and Materials

Required Texts

1. *A Practical Guide to Linux® Commands, Editors, and Shell Programming*, Third Edition. Mark G. Sobell. Prentice Hall. © 2012. ISBN-10: 0-13-308504-X. ISBN-13: 9780133085044
   Sobell example & source code: http://www.sobell.com/CR3/
   (Available online & free via UTD Library => eBook => Safari) This book is referred as [**Sobell**].

2. *Advanced Programming in the UNIX® Environment*, 3e. W. Richard Stevens and Stephen A. Rago. Addison-Wesley. © 2013. ISBN-10: 0-321-63773-9. ISBN-13: 9780321637734
   APUE source code: http://www.apuebook.com/code3e.html

(Available online & free via UTD Library => eBook => Safari) This book is referred as [**APUE**].

3. *Computer Systems: A Programmer's Perspective, Third Edition*. Randal E. Bryant and David R. O'Hallaron. © 2016 Pearson. This book is referred as [**CSAPP**]

Required Materials
Required Materials and/or Recommended Books

1. *Starting Out with C++ From Control Structures through Objects (with Access)* 8th edition. By Gaddis. ISBN-10: 0133796337 • ISBN-13: 9780133796339. (7th edition is OK, 0132576252)
   (This is the textbook for your cs1336 and cs1337 courses. Review ch09-16).
   This book is referred as [**Gaddis**]

2. *The C Programming Language, 2ed*. by Dennis M. Ritchie and Brian W. Kernighan. © 1988 Prentice Hall. ISBN: 9780133086249.
   (Available online & free via UTD Library => eBook => Safari). This book is referred as [**Cprog**].

3. *The Linux Programming Interface.* Michael Kerrisk. © 2010 No Starch Press. ISBN 978-1-59327-220-3
   (Available online & free via UTD Library => eBook => Safari). This book is referred as [**LPI**].

4. *Introducing Python*. Bill Lubanovic. © 2014 O'Reilly Media, Inc. ISBN-13: 978-1-4493-5936-2
   (Available online & free via UTD Library => eBook => Safari). This book is referred as [**Python**].

5. *Unix® and Linux® System Administration Handbook*, Fourth Edition, Video Enhanced Edition. by Evi Nemeth; Garth Snyder; Trent R. Hein; Ben Whaley. © 2010 Prentice Hall. ISBN-10: 0-13-148005-7. ISBN-13: 978-0-13-148005-6
   (Available online & free via UTD Library => eBook => Safari). This book is referred as [**Handbook**].

6. *The Sockets Networking API: UNIX® Network Programming*. Vol 1, 3ed. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff. © 2003 Addison-Wesley Professional. ISBN-10: 0-13-141155-1. ISBN-13: 978-0-13-141155-5. Source code: http://www.unpbook.com/
   (Available online & free via UTD Library => eBook => Safari) This book is referred as [**Network**].

7. *C++ How to Program*, 10/e. by Paul Deitel and Harvey Deitel. © 2016 Pearson. ISBN-13: 978-0-13-444823-7. ISBN-10: 0-13-444823-5
   (Available online & free via UTD Library => eBook => Safari). This book is referred as [**Deitel**].

8. *C++ Programming Language.* 4/e. Stroustrup ©2014 Addison-Wesley ISBN-10: 0321958322. ISBN-13: 9780321992789. (Available online & free via UTD Library => eBook => Safari)

9. *The C++ Programming Language, 4ed.* Bjarne Stroustrup. © 2013 Addison-Wesley Professional. ISBN-13: 978-0-321-56384-2. (Available online & free via UTD Library => eBook => Safari)

10. *A Tour of C++.* Bjarne Stroustrup. © 2013 Addison-Wesley Professional. ISBN-13: 978032195831. (Available online & free via UTD Library => eBook => Safari)

11. *C for Programmers with an Introduction to C11.* Harvey Deitel and Paul Deitel. © 2013 Prentice Hall. ISBN-10: 0-13-346206-4. ISBN-13: 978-0-13-346206-7
    (Available online & free via UTD Library => eBook => Safari)

12. *21st Century C, 2ed.* Ben Klemens. © 2014 O'Reilly Media, Inc. ISBN-13: 978-1-4919-0389-6
    (Available online & free via UTD Library => eBook => Safari)

13. *Intermediate C Programming.* Yung-Hsiang Lu. © 2015 CRC Press. ISBN 978-1-4987-1163-0.
    (Available online & free via UTD Library => eBook => Safari)

14. *Using SQLite.* Jay A. Kreibich. © 2010 O'Reilly Media, Inc. ISBN-13: 978-0-596-52118-9
(Available online & free via UTD Library => eBook => Safari)

15. *flex & bison.* John Levine. © 2009 O'Reilly Media, Inc. ISBN 9780596805418
(Available online & free via UTD Library => eBook => Safari) This book is referred as [**FlexBison**].

16. *Unix Systems Programming: Communication, Concurrency, and Threads.* Kay A. Robbins; Steven Robbins. © 2003 Prentice Hall. ISBN-10: 0-13-042411-0. ISBN-13: 978-0-13-042411-2
(Available online & free via UTD Library => eBook => Safari) This book is referred as [**USP**].

Online Resource and Web Sites

Sobell source code: http://www.sobell.com/CR3/
APUE source code: http://www.apuebook.com/code3e.html
Computer Systems: http://www.cs.cmu.edu/afs/cs/academic/class/15213-f15/www/schedule.html
Unix Network Programming - source code: http://www.unpbook.com/
Unix Systems Programming - http://usp.cs.utsa.edu/usp/

C++ language tutorial http://www.cplusplus.com/files/tutorial.pdf
C++ tutorial http://www.learncpp.com/
C++ reference: http://cppreference.com

MobaXterm: http://mobaxterm.mobatek.net/
Putty http://www.putty.org/
Filezilla https://filezilla-project.org/

Unix/Linux commands: https://kb.iu.edu/d/afsk
Linux Shell and Commands: http://vic.gedris.org/Manual-ShellIntro/1.2/ShellIntro.pdf
POSIX Thread Programming Tutorial. https://computing.llnl.gov/tutorials/pthreads/
Thread Programming http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html
Python.org https://www.python.org/
Sqlite3 https://www.sqlite.org/

Suggested Course Materials

Suggested Readings/Texts

Suggested Materials

Textbooks and some other bookstore materials can be ordered online through Off-Campus Books http://www.offcampusbooks.com or the UT Dallas Bookstore http://www.bkstr.com/texasatdallasstore/home. They are also available in stock at both bookstores.

Technical Requirements

In addition to a confident level of computer and Internet literacy, certain minimum technical requirements must be met to enable a successful learning experience. Please review the important technical requirements http://www.utdallas.edu/elearning/students/getting-started.html#techreqs on the Getting Started with eLearning webpage http://www.utdallas.edu/elearning/students/getting-started.html.

Course Access and Navigation

The course can be accessed using the UT Dallas NetID account at: https://elearning.utdallas.edu.

Please see the course access and navigation http://www.utdallas.edu/elearning/students/getting-started.html#courseaccessandnav section of the site for more information.

To become familiar with the eLearning tool, please see the Student eLearning Tutorials http://www.utdallas.edu/elearning/students/eLearningTutorialsStudents.html.

UT Dallas provides eLearning technical support 24 hours a day/7 days a week. The eLearning Support Center http://www.utdallas.edu/elearninghelp services include a toll free telephone number for immediate assistance (1-866-588-3192), email request service, and an online chat service.

Communication
This course utilizes online tools for interaction and communication. Some external communication tools such as regular email and a web conferencing tool may also be used during the semester. For more details, please visit the eLearning Tutorials webpage http://www.utdallas.edu/elearning/students/eLearningTutorialsStudents.html for video demonstrations on eLearning tools.

Student emails and discussion board messages will be answered within 3 working days under normal circumstances.

Distance Learning Student Resources
Online students have access to resources including the McDermott Library, Academic Advising, The Office of Student AccessAbility, and many others. Please see the eLearning Current Students page http://www.utdallas.edu/elearning/students/cstudents.htm for details.

Server Unavailability or Other Technical Difficulties
The University is committed to providing a reliable learning management system to all users. However, in the event of any unexpected server outage or any unusual technical difficulty which prevents students from completing a time sensitive assessment activity, the instructor will provide an appropriate accommodation based on the situation. Students should immediately report any problems to the instructor and also contact the online eLearning Help Desk http://www.utdallas.edu/elearninghelp. The instructor and the eLearning Help Desk will work with the student to resolve any issues at the earliest possible time.

Assignments & Academic Calendar*

***Note:** The descriptions and timelines contained in this syllabus are subject to change at the discretion of the Professor.

Dates	UNIT in Week #	TOPIC/LECTURE	READING*	ASSESSMENT Weekly Activity items (Activity, Quiz, Essay, and 2 Posts - See elearning for detail), Assignment, Test	DUE DATE
	0	Orientation & Prerequisite Form		Week00 Activity (to sign and upload the completed prerequisite form PDF to elearning)	
8/20 M	1	Syllabus & Course Introduction Unix/Linux Introduction & Commands • Connect to cs1 from your laptop (with MobaXterm or SSH) • Simple C programming (Hello World) • Simple file-editing with vi	Read Sobell Ch1-3 APUE01	Week01 Activity (to download, install and try mobaXterm or ssh or putty to connect cs1, etc.) See Week01 Activity folder in elearning for detail	8/25 Sat
8/27 M	2	Unix, Linux Commands (Advanced) File Systems (Sobell Ch4) Shell (Sobell Ch5) Editors (Sobell Ch6) Makefile	Read Sobell Ch4-6 Read APUE02	Week02 Activity	9/01 Sat
9/03 M	3	Bourne Again Shell – bash (Sobell 8, 10) Shell Script Programming with bash	Read Sobell Ch8, 10 APUE03	Week03 Activity	9/08 Sat
9/10 M	4	MySQL (Sobell Ch13) sqlite3 prog	Read Sobell Ch13 APUE04	Week04 Activity	9/15 Sat
9/17 M	5	Python prog (Sobell Ch12) Python (OpenCV) Unix/Linux System Prog & API	Read Sobell Ch12 APUE05	Week05 Activity Assignment #1 9/11 Monday Noon	9/22 Sat A1 9/17 M
9/24 M	6	Unix File Systems and IO, and API	Read APUE 07 Review APUE03-04	Week06 Activity Test1 9/25 T	9/29 Sat T1 9/25
10/01 M	7	Unix File Systems and IO, and API	Read APUE 08 Review APUE03-04	Week07 Activity	10/06 Sat

10/08 M	8	Process	Read APUE09	Week08 Activity	10/13 Sat
10/15 M	9	Process	Read APUE 10	Week09 Activity Assignment2 10/15 M Noon	10/20 Sat A2 10/15
10/22 M	10	Shell Signal	Read APUE 11 Review APUE07-09	Week10 Activity Test2 10/23 T	10/27 Sat T2 10/23
10/29 M	11	Threads	Read APUE 12	Week11 Activity	11/03 Sat
11/05 M	12	Interprocess Communication (IPC)	Review APUE 11.6 Read APUE 15	Week12 Activity	11/10 Sat
11/12 M	13	Socket Programming	Read APUE 16	Week13 Activity Assignment3 11/06 M Noon	11/17 Sat A3 11/06
11/19 M	14	Fall Break			
11/26 M	15	Socket Prog - Concurrent Server Advanced Topics	Read APUE16	Week15 Activity Test3 11/27 T	12/01 Sat T3 11/27
12/03 M	16	Advanced Topics 12/06 W Last Day of Class	Flex & Bison VCS (Git, Github, Gitlab)	Week16 Activity Assignment #4 12/03 M Noon (No late submission accepted)	12/08 W A4 12/03
12/10 M	17	Final Exam Week			
		Final Grade Due 12/20 Th			

For Required Reading.

APUE - *Advanced Programming in the UNIX® Environment*, 3e. W. Richard Stevens and Stephen A. Rago. Addison-Wesley. © 2013. ISBN-10: 0-321-63773-9. ISBN-13: 9780321637734
(Available online & free via UTD Library => eBook => Safari)

Sobell – *A Practical Guide to Linux® Commands, Editors, and Shell Programming*, Third Edition. Mark G. Sobell. Prentice Hall. © 2012. ISBN-10: 0-13-308504-X. ISBN-13: 9780133085044
(Available online & free via UTD Library => eBook => Safari)

Proctored Final Exam Procedures
If your course has a proctored exam requirement, please see the UTD Student Success Center – Testing Center website http://www.utdallas.edu/studentsuccess/testing-center/ to make arrangements.

Note: All tests are scheduled and held at Test Center (Student Success Center). See the detail below to make your seat reservation and/or to arrange any makeup test, etc.

Grading Policy
Letter grades will be assigned as follows:

97-100	A+	93-96	A	90-92	A-
87-89	B+	83-86	B	80-82	B-
77-79	C+	73-76	C	70-72	C-
67-69	D+	63-66	D	60-62	D-
Below 60	F				

Note: Each range shown above is inclusive and without any rounding-off. For example, 93-97 for grade A is for the score falling in the range between 93.000 and 96.999. The grade of 92.999 is for A-.

Note: In elearning, "Running" total in your gradebook shows the current weighted grade based on your graded work only based on what you have submitted and graded. For example, if you have done only Test1, Assignment1, Weekly postings so far (but you have missed Test2 and missed Assignment2 totally), current total grade will be based on only those entries that you have submitted and done.

45% for 3 Tests. 15% for each test. Each test will be taken at Testing Center (Student Assessment Center, McDermott Library 1st floor) for 2-hour examination. Time of Test will be announced later in elearning. Each student should make a seat reservation prior to each test (as soon as possible). All exams are closed book and closed notes. Exams will focus more on concepts and less on details. Necessary documentation will be provided to avoid the need for memorization as much as possible. We will likely take all the tests in the testing center as scheduled. You can expect to see a few coding/analysis questions, a few short answer questions and a few multiple-choice questions in each test. Instructor is responsible for grading all the tests.

Any make-up tests will be arranged and scheduled during the same week (usually Tuesdays prior to the actual test date) at the discretion of the instructor. There should be a valid reason for scheduling make-up tests & they need to be coordinated with the instructor, 1-2 weeks prior to the test date except for serious medical condition (with Doctor's or Hospital's certificate will be required as a valid proof. Without it, there will be 15% penalty for any makeup test after the scheduled test date). It is unlikely that curving will be used to boost the final grades. If the instructor decides to do it, only the test scores will be boosted, but the tests' contribution will be clipped at 60%. In other words, curving will NOT make up for the points lost in all other assignments. So, it is extremely important to complete them in timely manner.

40% for 4 Assignments (projects) contributing 10% each, Due Monday 12pm Noon. No late submission is accepted. You can ask for clarifications and help in the weekly forum. If you need help with your code, it is ok to post 1 or 2 lines of code, but do not post your full program - email it to TA or professor instead. You are expected to start working on them as soon as they are posted. Do not expect us to rescue you on the day of submission. I encourage everyone to submit the projects 1 or 2 days early. You can upload it again but

the last submission will be graded. [Do not wait until the last minute to submit it. I do understand things happen and occasionally as you may not be able to submit projects on time.]

No Late submission is accepted. My advice is to submit whatever you have done (your best effort) before the due and/or by the due, to seek for any further discretion and/or consideration.

All these assignments/projects should be done in Unix, Linux or Mac, and you will hand-in your projects directly in Linux.

Submit your assignment through elearning (Assignments folder). More details on Assignment & Submission steps will be given with eLearning. For each assignment, TA may schedule a demo and you are required to schedule your demo with TA (for 5-10 minutes) and do your demo to TA. If you have any conflict for the demo schedule, you may do the demo to the instructor (and/or you record your demo in a video format using your webcam using WebEx (from Cisco and all UTD students should have access to this tool), and upload it to elearning as a part of your assignment, and notify TA.

Warning. To get A- or above (in letter grade), student should complete and submit all of the assignments and get over 60% for each assignment. To get B- or above, student should complete and submit at least 50% of the assignments (2 out of 4 Assignments), and get over 50% or more for each assignment submitted.

An instructor who believes a student has committed an act of **plagiarism** should take appropriate action, which includes the issuing of a "penalty grade" (that is, F for the course) for academic dishonesty. For any "minor" plagiarism charge, the maximum letter grade for the course would be B+ or lower.

15% for Weekly Activity (Activity, Quiz, Essay, Post [Discussion Board]) (including online quiz) is available and posted by Monday & will be due by Saturday midnight (11:59pm) each week. It will be a small programming exercise or tryout (e.g., to write and run a simple "Hello world" program, to try Linux commands or sample programs provided, to install a tool to try it) in most weeks. It can also be a quiz (online and open-book) or some other meaningful activity as well. It will vary every week. Each weekly activity and its score may vary case by case. Late submissions are NOT accepted for weekly activities and quizzes. Note: Weekly quiz will provide a good snapshot, an excellent opportunity to review, and for a preparation for each test. Late submissions are NOT accepted for weekly activity or quiz.

Weekly Postings. 2 meaningful and relevant posts are required every week in weekly discussion forums. This is extremely crucial component of a true online course. No non-sense and no trivial comment. One-liners saying "Thanks!" ("Weather is bad" or "I got it" or "I do not know" or "very good" etc.) will not be counted as a valid posting or participation. Keep your posting very relevant and valuable to you and your classmates, and to the course work and activity of the week. Your post can be a good question, meaningful response to another student's question, interesting observation, etc. For a question, you should do your own homework for your question and share your findings. If you use an external source (web page, link, Youtube video, etc.), you should provide a reference or a link of the source with a good overview or summary in your own wording and reflection. Do not post any offending or destructive content. Do not post any overwhelming contents (e.g., to copy and paste big image or images, or very long text content, or using "big" fonts) but you should attach a file as you need. In simple words, each post should value to the course. Instructor (TA or Grader) will grade the weekly forum and determine the value of each post - instructor's decision is final. First post should be submitted latest by Wednesday midnight and 2nd post should be completed latest by Saturday midnight, otherwise respective posts won't receive any grade. It is possible for someone to be a silent observer in on-ground course and still manage to get the final grade of A. It is impossible to do it in online course. Reasonable progress towards the expected answer or learning will get 1 point & perfect or near-perfect submissions will get 2 points. Late submissions are NOT accepted for weekly posts. After the due, the weekly post will not be available.

Course Policies

Instructor is responsible for grading all the tests & weekly participation. TA will be responsible for grading projects and weekly assignments. So, contact the TA directly for any grading related discrepancies for programs. It is not possible to give a detailed feedback for each project/weekly assignment/test question due to large # of students in our classes. If you need more details/clarification, you are encouraged to meet the TA/instructor during office hours & get personal attention. Do not rely on email alone to get the full response. If you are stuck with your assignment, it is better to turn in what you have and send us email. We will revise your submission and give some guidance. Your next submission will override the previous submission - TA will always grade the latest submission for each project. You can use email to get help for weekly assignments. Include the detailed problem description & applicable error messages, zip all your source files and include it with your email too. Do not just say "my program does not work" and expect us to figure out everything - you need to help us to help you efficiently. We expect to complete grading assignments (projects), weekly activities or quizzes, and tests in a week or so. However, when the schedule gets too busy, it can be as long as 2 weeks before the grades are assigned. It is the students' responsibility to review the grade details when they become available and follow up for clarifications if needed.

Attendance. For in-class course (and especially for online course: it will be tracked and assessed via your weekly activity, quiz, essay, and weekly postings, counted as your weekly class-participation via elearning), Attendance Rule & Policy: Please note that if you miss any lectures beyond the 1st week, then automatic actions kick in: (1) Missing the next lecture in the 2nd week will result in an automatic drop of one grade from your final course grade. (2) Missing the entire 2nd week of lecture(s) is an automatic F in the course. So if you are going to miss more than one week of classes (ideally, you should not miss any lecture, but sometimes people switch courses during the first week), then you should not be in the course and you should drop out. Further you should plan to be here for Final Examination Week, as it will be scheduled for this course.

Course Policies

Make-up exams

Any make-up tests will be scheduled during the same week (usually Tuesdays prior to the actual test date) at the discretion of the instructor. There should be a valid reason for scheduling make-up tests & they need to be coordinated with the instructor, 1-2 weeks prior to the test date except for serious medical condition (with Doctor's or Hospital's certificate will be required as a valid proof.) Without any valid reason, there will be 15% penalty for any makeup test after the scheduled test date.

Extra Credit

Late Work

No late submission of any work is accepted unless stated otherwise.

Special Assignments

Class Participation

For all in-class courses, the attendance is required for each class, tests and demo.
For online-course, each weekly activity will be counted as your attendance.

Classroom Citizenship

Comet Creed

This creed was voted on by the UT Dallas student body in 2014. It is a standard that Comets choose to live by and encourage others to do the same:

"As a Comet, I pledge honesty, integrity, and service in all that I do."

UT Dallas Syllabus Policies and Procedures

The information contained in the following link constitutes the University's policies and procedures segment of the course syllabus.

Please go to http://go.utdallas.edu/syllabus-policies for these policies.

The descriptions and timelines contained in this syllabus are subject to change at the discretion of the Professor.