| | | |
|---|---|---|
| UTD | **Course** | CS/SE 3377.502<br>CS/SE 3377.006 |
| | **Professor** | Dr. Mohamed Amine Belkoura |
| | **Term** | Fall 2024 |
| | **Meetings** | 4:00-5:15pm (section 006), Tu/Th **ECSS 2.305**<br>5:30-6:45pm (section 502), Tu/Th **ECSS 2.306** |

**Professor's Contact Information**

| | |
|---|---|
| **Office Location** | No Office |
| **Email Address** | mxb135330@utdallas.edu, Subject: [CS3377 + section] + subject |
| **Office Hours** | through Microsoft Teams, send private message. |

**General Course Information**

| | |
|---|---|
| **Pre-requisites** | CE 2336 or CS 2336 or TE 2336 |
| **Course Description** | CS3377**: C/C++ Programming in a UNIX Environment** (3 semester credit hours) C/C++ Programming in a UNIX Environment (3 semester credit hours) Advanced programming techniques utilizing procedural and object oriented programming in a UNIX environment. Topics include basic UNIX concepts, file input and output, implementation of strings, and dynamic memory allocation/management. Design and implementation of a comprehensive programming project is required. 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 |
| **Learning Outcomes** | After successful completion of this course, the student should be able to:<br>- Ability to use Unix/Linux operating system (command line interface, shell scripting, regular expression)<br>- Ability to use Unix/Linux programming environment and development tools (gcc/cc/c++ compiler, vi/vim editor, etc.)<br>Ability to program with Unix/Linux processes, threads, and interprocess communication facilities.<br>- Ability to program with Unix/Linux file system, file input and output, and redirection.<br>- Ability to develop programs for network environment (client-server model, socket programming, and cloud computing) |
| **Required Text** | 1. *A Practical Guide to Linux® Commands, Editors, and Shell Programming*, Fourth 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**].<br><br>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**]. |

| | |
|---|---|
| **Supplemental Text** | 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**]<br><br>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**].<br><br>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**].<br><br>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**].<br><br>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**].<br><br>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**].<br><br>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**].<br><br>8. *C++ Programming Language*. 4/e. Stroustrup ©2014 Addison-Wesley ISBN-10: 0321958322. ISBN-13: 9780321992789. (Available online & free via UTD Library => eBook => Safari)<br><br>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)<br><br>10. *A Tour of C++*. Bjarne Stroustrup. © 2013 Addison-Wesley Professional. ISBN-13: 978032195831. (Available online & free via UTD Library => eBook => Safari)<br><br>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)<br><br>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)<br><br>13. *Intermediate C Programming*. Yung-Hsiang Lu. © 2015 CRC Press. ISBN 978-1-4987-1163-0. (Available online & free via UTD Library => eBook => |

| | |
|---|---|
| **Additional Online Resource and Web Sites** | Sobell source code: http://www.sobell.com/CR3/<br>APUE source code: http://www.apuebook.com/code3e.html<br>Computer Systems: http://www.cs.cmu.edu/afs/cs/academic/class/15213-f15/www/schedule.html<br>Unix Network Programming - source code: http://www.unpbook.com/<br>Unix Systems Programming - http://usp.cs.utsa.edu/usp/<br><br>C++ language tutorial: http://www.cplusplus.com/files/tutorial.pdf<br>C++ tutorial http://www.learncpp.com/<br>C++ reference: http://cppreference.com<br><br>MobaXterm: http://mobaxterm.mobatek.net/<br>Putty http://www.putty.org/<br>Filezilla https://filezilla-project.org/<br><br>Unix/Linux commands: https://kb.iu.edu/d/afsk<br>Linux Shell and Commands: http://vic.gedris.org/Manual-ShellIntro/1.2/ShellIntro.pdf<br>POSIX Thread Programming Tutorial. https://computing.llnl.gov/tutorials/pthreads/<br>Thread Programming http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.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.<br><br>To become familiar with the eLearning tool, please see the Student eLearning Tutorials http://www.utdallas.edu/elearning/students/eLearningTutorialsStudents.html.<br>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.<br><br>Student emails and discussion board messages will be answered within 3 working days under normal circumstances. |

| | |
|---|---|
| **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. |

**Important Dates***

| | |
|---|---|
| **08/20 Tuesday** | First Day of Class |
| **09/25-09/26** | Exam1 (registration required) |
| **10/30-10-31** | Exam2 (registration required) |
| **11/25 - 12/01** | NO CLASSES (ThanksGiving) |
| **12/05** | Final exam (in class) |

* Note: The dates here are tentatively assigned and are subject to change as needed.

**Course Grading**

| | | |
|---|---|---|
| **Grading Criteria** | Assignments 40%<br>Homework/Quiz 10%<br>Participation 5%<br>2 Tests (15+15) 30%<br>Final exam 15% | 97-100 A+<br>93-96 A<br>90-92 A-<br>87-89 B+<br>83-86 B<br>80-82 B-<br>77-79 C+<br>73-76 C<br>70-72 C-<br>67-69 D+<br>63-66 D<br>60-62 D-<br>Below 60 F |
| **Make-up Exams** | Not allowed | |
| **Late Work** | Not allowed | |
| **All other policies** | Please visit http://go.utdallas.edu/syllabus-policies for other policies | |

**Tentative Schedule**

| Week | Topic |
|---|---|
| 1 | Introduction to CS2336, syllabus review |
| 2 | Unix Basics |
| 3 | Unix File System |
| 4 | VI editor |
| 5 | Building C/C++ in Unix/Linux |
| 6 | **Exam1** |
| 7 | Bash shell |
| 8 | Regular expressions |
| 9 | Pointers in C/C++ |
| 10 | Input/output |
| 11 | **Exam 2** |
| 11 | Source control |
| 12 | Multiprocessing |
| 13 | Thread programing |
| 14 | Socket programing in C |
| 15 | **Final Exam** |

**Course Policies**

1. **30% for 2 Tests + 15% for final exam.** 15% for each test. Test 1 will cover first part of semester instruction, and 2nd exam for the second part of the semester. There will be A **FINAL** test in the last week of the semester, **NOT final examination week.** Time of Test will be announced later in elearning. All exams are closed book and closed notes. Exams will on concepts as well as details (commands, scripts…).. 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 within 1-2 weeks of the beginning of the semester, except for serious medical condition (with Doctor's or Hospital's certificate will be required as a valid proof. Without it, there will be no makeup test). 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.

2. **40% for all assignments/projects** (number to be determined), No late submission is accepted. You can ask for clarifications through elearning. 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.

Submit your assignment through eLearning (Assignments folder). More details on Assignment & Submission steps will be given in 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.

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.

3. 10**% for Homework and quizzes:** An example homework might consist of 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 or some other meaningful activity as well. Late submissions are NOT accepted

4. **5% for Participation/Attendance:** Participation is tracked through class discussion Participation will be used to also measure attendance. Please note that if you miss any weekly activity 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, then you should not be in the course and you should drop out

**Grading Policies**

Instructor is responsible for grading all the tests. TA will be responsible for grading projects and 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.