

CSE 1320

Week of 01/14/2019

Instructor : Donna French

Syllabus

UNIVERSITY OF
TEXAS
ARLINGTON | Blackboard

My Blackboard Students Library

2192-CSE-1320-002-INTERMEDIATE-PROGRAMMING--2019-Spring Syllabus

Syllabus

Build Content Assessments Tools Partner Content

CSE 1320 - Spring 2019

Enabled: Statistics Tracking

Attached Files: [Syllabus - CSE 1320 - Spring 2019.pdf](#) (761.81 KB)

UTA Honor Code

Attached Files: [UTAHonorCodeAcadIntegStmt.pdf](#) (488.343 KB)

Please read this document. Print the second page, fill it out and sign it. Please turn it in to me at the beginning of class on January 17th.

For the semester fill in the blank, please put "Spring 2019 CSE 1320".

Syllabus

Course Materials

Calendar

Crash Course

Homework

Coding Assignments

UTA Email

Syllabus

CSE 1320 – Intermediate Programming

Spring 2019

Instructor Donna French

Office Number ERB 338

Office Phone I don't have a phone in my office, but in case of an emergency you can call the CSE department at 817-272-3785.

Email Address donna.french@uta.edu (best way to contact me)

Faculty Profile <https://mentis.uta.edu/explore/profile/donna-french>

Office Hours Monday, Wednesday, Friday 10:30 pm – 12:30 pm (or anytime my door is open)

2:30 pm – 4:00 pm (or anytime my door is open)

Tuesday, Thursday 10:00 am – Noon (or anytime my door is open)

Section 008 Monday, Wednesday, Friday 9:00 AM – 9:50 AM ERB 130

003 Monday, Wednesday, Friday 1:00 PM – 1:50 PM COBA 251

002 Tuesday, Thursday 12:30 PM – 1:50 PM ERB 131

Course Description: Programming concepts beyond basic control and data structures. Emphasis is given to data structures including linked-lists and trees as well as modular design consistent with software engineering principles.

Prerequisite: C or better in CSE 1310 or C or better in CSE 1312, and C or better in (or concurrent enrollment in) (MATH 1421, MATH 1426, MATH 2425, MATH 2326, MATH 3330, HONR-SC 1426, or HONR-SC 2425) and C or better in CSE 1105 (or concurrent enrollment).

Course Objectives:

- Introduction to the C programming language
- Exposure to basic data structures
- Learn to use the Linux operating system

Textbook: C By Discovery

Foster & Foster

4th Edition

ISBN 9781576761700

Syllabus

Homework Policy: Programming is learned by doing – not just by reading about it or listening to someone talk about it. How well would you play a sport or a musical instrument if you only read about how to play or listened to someone lecture about how to play? There will be coding assignments and homework assignments almost every week. You will not be able to pass this class with a C or better unless you do the coding assignments. The homework assignments are to reinforce the in class presentations and will serve as your study guides for the exams.

Since all assignments/code will be submitted via Blackboard, you will have 24 hours after the due date/time to submit your work but it will incur a 50% penalty. PLEASE remember that 50 points out of 100 is better than ZERO.

Please note – Coding Assignments that do not compile or compile with ANY warnings will be assigned a grade of 0 automatically. No partial credit will be given for code that does not cleanly compile. Code must run in order to be tested/graded.

While I do encourage students to work together on understanding Coding Assignments, I expect every student to do their own work and turn in their own code. Coding Assignments are checked for similarity – any student's code that is determined to be too similar to another student's code submission will be assigned a 0 for the first offense and will be referred to the Office of Student Conduct for any subsequent incidents. This policy will be applied to all students involved – does not matter if you are copying someone else or allowing someone else to copy you.

Any assignment from this class that is found posted on the web in any format will be voided for all sections and a new coding assignment will be assigned to all students unless the student who attempted to cheat via the web is identified.

Honor Code

The following is an excerpt from the College of Engineering's statement on Ethics, Professionalism, and Conduct of Engineering Students. The notes are modifications appropriate for this Computer Science and Engineering course. Read the statement carefully, print on ONE page, sign it, and return it to your instructor.

Statement on Ethics, Professionalism, and Conduct for Engineering Students
College of Engineering
The University of Texas at Arlington
<http://www.uta.edu/engineering/current-students/academic-honesty.php>

The College cannot and will not tolerate any form of academic dishonesty by its students. This includes, but is not limited to cheating on examination, plagiarism, or collusion (explained below).

Definitions:

- A. **Cheating on an examination** includes:
 - 1. Copying from another's paper, any means of communication with another during an examination, giving aid to or receiving aid from another during an examination;
 - 2. Using any material during an examination that is unauthorized by the proctor;
 - 3. Taking or attempting to take an examination for another student or allowing another student to take an examination for oneself;
 - 4. Using, obtaining, or attempting to obtain by any means the whole or any part of an examination.
- B. **Plagiarism** is the unacknowledged incorporation of another's work into work submitted for credit. This includes use of material found on the Internet except as authorized by the instructor.
- C. **Collusion** is the unauthorized collaboration of another in preparing work to be submitted for credit.
- D. Other types of **academic dishonesty** include allowing another person to use one's account, using the account for any purpose than violates university policy, misappropriation of property, or aiding someone else, and similar offenses.

Notes:

- 1. The use of the source code of another person's program, even if modified, is considered plagiarism.
- 2. Allowing another person to use your source code, even temporarily, is considered plagiarism.
- 3. In this class, the specific exceptions given below are *not* considered academic dishonest acts:
 - A. Discussion of the algorithm and general program design with another student is not a problem.
 - B. Giving and receiving aid in debugging is not a problem.
 - C. Discussion and comparison of program output is not a problem.
- 4. In this class, the penalty assessed for cheating is failing the assignment. The student will be referred to the [Handbook of Operating Procedures](#) and the [Regents' Rules and Regulation](#) (www.uta.edu/conduct) for further information.
- 5. You may be entitled to know what information is contained in your student record. You may review and have UTA correct this information according to the [UT System BPM #32](#). The law is found in sections 552.021, 552.023 and 559.004 of the Texas Education Code.

I have read and I understand the above statement of the College's evaluation process, taking of examinations, and the submission of written, videoed/filmed, and submitted papers, computer programs, exams, homework, projects, and other submissions may be computer evaluated for plagiarism of words, figures, program code.

Student's signature: _____

Student's name (printed): _____

Student MyMav login ID: _____ UTA ID number: **100**

Semester _____ Date _____

The following is an excerpt from the College of Engineering's statement on Ethics, Professionalism, and Conduct of Engineering Students. The notes are modifications appropriate for this Computer Science and Engineering course. Read the statement carefully, print on ONE page, sign it, and return it to your instructor.

Statement on Ethics, Professionalism, and Conduct for Engineering Students

College of Engineering
The University of Texas at Arlington

<http://www.uta.edu/engineering/current-students/academic-honesty.php>

The College cannot and will not tolerate any form of academic dishonesty by its students. This includes, but is not limited to cheating on examination, plagiarism, or collusion (explained below).

Definitions:

- A. **Cheating on an examination** includes:
 - 1. Copying from another's paper, any means of communication with another during an examination, giving aid to or receiving aid from another during an examination;
 - 2. Using any material during an examination that is unauthorized by the proctor;
 - 3. Taking or attempting to take an examination for another student or allowing another student to take an examination for oneself;
 - 4. Using, obtaining, or attempting to obtain by any means the whole or any part of an examination.
- B. **Plagiarism** is the unacknowledged incorporation of another's work into work submitted for credit. This includes use of material found on the Internet except as authorized by the instructor.
- C. **Collusion** is the unauthorized collaboration of another in preparing work to be submitted for credit.
- D. Other types of **academic dishonesty** include allowing another person to use one's account, using the account for any purpose than violates university policy, misappropriation of property, aiding someone else, and similar offenses.

Notes:

- 1. The use of the source code of another person's program, even if modified, is considered plagiarism.
- 2. Allowing another person to use your source code, even temporarily, is considered plagiarism.
- 3. In this class, the specific exceptions given below are *not* considered academic dishonest acts:
 - A. Discussion of the algorithm and general program design with another student is not a problem.
 - B. Giving and receiving aid in debugging is not a problem.
 - C. Discussion and comparison of program output is not a problem.
- 4. In this class, the penalty assessed for cheating is failing the assignment. The student will be referred to the [Handbook of Operating Procedures](#) and the [Regents' Rules and Regulation](#) (www.uta.edu/conduct) for further information.
- 5. You may be entitled to know what information is contained in your student record. You may review and have UTA correct this information according to the [UT System BPM #32](#). The law is found in sections 552.021, 552.023 and 559.004 of the Texas Education Code.

I have read and I understand the above statement of the College's evaluation process, taking of examinations, and the submission of written, videoed/filmed, and submitted papers, computer programs, exams, homework, projects, and other submissions may be computer evaluated for plagiarism of words, figures, program code.

Student's signature: _____

Student's name (printed): _____

Student MyMav login ID: _____ UTA ID number: **100**

Semester _____ Date _____

Syllabus

Grading Policy: Letter grades will be assigned as follows: 90-100 = A, 80-89 = B, 70-79 = C, 60-69 = D, 0-59 = F.

| | |
|-----------------------|--|
| Exam 1 | 10% |
| Exam 2 | 10% |
| Final Exam | 30% (Departmental Final) – must earn at least 70% in order to pass class |
| Homework/Crash Course | 10% |
| Code | 20% |
| In class quizzes | 20% |

No make-up exams will be given except for extenuating circumstances beyond the student's control (in the instructor's opinion). Poor planning or forgetfulness on your part won't be considered an emergency

Syllabus

Attendance: At The University of Texas at Arlington, taking attendance is not required but attendance is a critical indicator in student success. As the instructor of this section, I will not specifically take attendance but a significant portion of your final grade will be earned in class; therefore, if you do not attend regularly, your grade will suffer. If you have notified me BEFORE missing class that you will be missing class, then you will be allowed an opportunity to make up any quizzes executed during that class period. If you do not notify me BEFORE missing class, any quiz given during the missed class will be an automatic 0.

Syllabus

Important Dates

| | |
|---|---|
| Monday, January 14 th | First day of class |
| Monday, January 21 st | No class – MLK Day Holiday |
| After completion of Arrays | Exam 1 |
| March 11 th – March 15 th | No class – Spring Break |
| After completion of Formatted I/O | Exam 2 |
| Friday, May 3 rd | Last day of classes at UTA |
| Monday, May 6 th | Departmental Final Exam (5:30 PM to 8:00 PM)* |

*All sections of CSE 1320 will take their final exam at this time, regardless of the time the class meets for lectures.

Homework

- Homework will be assigned on Mondays at noon and will be due the following Monday by midnight.
- Homework will usually consist of questions covering the previous week's lectures.
- Any homework submitted within 24 hours after the due date will incur a 50% penalty but 50% is better than 0%.
- Homework questions are taken directly from the slides, class discussion and coding assignments.
- Homework will appear in Blackboard as “Tests” in order to make use of Blackboard’s ability to automatically grade that type of submission.
- Completing and understanding the homework will prepare you for the exams.

Crash Course

- Crash Course video quizzes will be assigned on Mondays at noon and will be due the following Monday by midnight.
- Any quizzes submitted within 24 hours after the due date will incur a 50% penalty but 50% is better than 0%.
- Quiz questions are taken directly from the videos. I highly suggest you turn on Closed Captioning to get exact phrasing and spelling. Please note that using the transcript to answer quiz questions (rather than watching the video) is strongly discouraged because the transcripts are created by non-Crash Course individuals and are NOT always correct. I will not accept the transcript answer.
- Crash Course quizzes will appear in Blackboard as “Tests” in order to make use of Blackboard’s ability to automatically grade that type of submission.
- The purpose of the quiz is to verify that you have watched the video – not to test how much you already know; therefore, alternate terms will not be accepted even though they may be correct.
- This should be an easy 100% every time.

Coding Assignments

- Coding Assignments will be assigned on Mondays at noon and will be due the following Monday by midnight.
- Some of the assignments will build on the previous week's assignment. They will start out easy and will get progressively harder.
- Turn in early and ask me to review. Email me what you have and ask questions. Don't get stuck.
- Coding Assignments will be graded by your assigned GA. They will use the provided rubric to grade your programs. Review the rubric yourself before submitting your final version. **Code that does not compile or compiles with warnings on Omega will automatically receive a grade of 0.**
- Google is not your friend
 - Ask Google – find a thousand places to look
 - Ask your professor – find the right place to look

Exams

- Exams will be based on homework, coding assignments and in class discussion.
- Crash Course information will not be included on exams.
- If you complete and understand the homework and the coding assignments, you will be well prepared for the exams.

Departmental Final Exam

Departmental Final: The Final Exam grade is 30% of the total class score and both of the following criteria MUST be met to pass this class.

students are required to make 70% or above on the Departmental Final Exam

AND

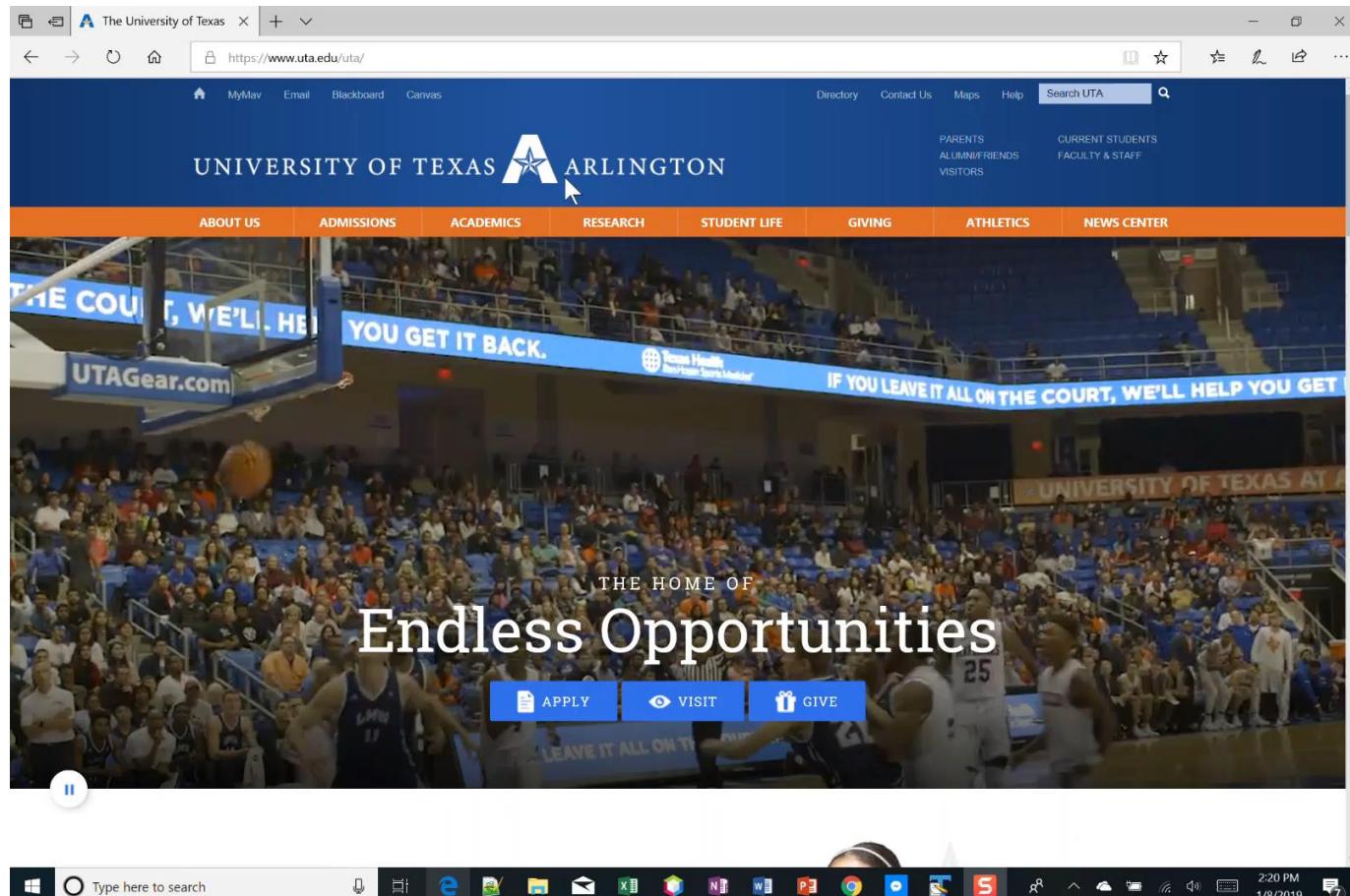
students are required to make 70% or above for the overall class score

In Class Quizzes

Quizzes given in class will make up 20% of your overall grade.

These quizzes will test your ability to write code. All of your exams will require handwritten code.

Blackboard



Blackboard Resource

The screenshot shows the Blackboard Course Materials page for the course 2192-CSE-1320-002-INTERMEDIATE-PROGRAMMING--2019-Spring. The left sidebar lists various course sections and tools. The 'Course Materials' section is highlighted with a red box. The main content area displays several folder icons with their respective contents:

- Textbook for CSE 1320**
- Omega FileZilla PuTTY Mac UNIX GDB**
- GDB**
- Videos From Class**
- Slides**
- Demo Programs from Class**

Two blue arrows point to the 'Slides' and 'Demo Programs from Class' folders with the following annotations:

- Slides**: Slides from lecture will be posted here.
- Demo Programs from Class**: Demo programs from lecture will be posted here.

On the right side of the page, there is an orange vertical bar labeled '24X7 Bb Support' with a question mark icon.

Notepad++

The screenshot shows the homepage of the Notepad++ website at <https://notepad-plus-plus.org/>. The page has a dark header with a lock icon and the URL. Below the header is a navigation bar with links for 'Apps' and 'Blog'. The main content area features a large orange sidebar on the left with a cartoon chameleon logo and the text 'Notepad++'. The sidebar contains links: 'home', 'download' (which is highlighted with a green oval and a red arrow pointing to it), 'news', 'features', 'resources', 'contribute', 'donate', 'community', 'contributors', and 'links'. At the bottom of the sidebar is a 'download' button. The main content area has two sections: 'News' and 'About'. The 'News' section lists recent releases from 2017 to 2018. The 'About' section provides a brief overview of Notepad++ as a free source code editor and its features. A large inset image in the 'About' section shows a screenshot of the Notepad++ application window displaying a C++ code editor with syntax highlighting.

Secure | https://notepad-plus-plus.org//

Apps Blog

more languages

News

- Notepad++ 7.5.8 released
Feb 24 2018
- Notepad++ 7.5.7 released
Jun 30 2018
- Notepad++ 7.5.6 : a message from outer space
Mar 19 2018
- Notepad++ 7.5.5 released
Feb 27 2018
- Notepad++ 7.5.4 released
Jan 01 2018
- Notepad++ 7.5.3 released
Dec 06 2017
- Notepad++ 7.5.2 released
Nov 28 2017
- Notepad++ 7.5.1 released
Aug 30 2017
- Notepad++ 7.5 released
Aug 16 2017
- Notepad++ 7.4.2 released
Jun 18 2017

About

Notepad++ is a free (as in "free speech" and also as in "free beer") source code editor and Notepad replacement that supports several languages. Running in the MS Windows environment, its use is governed by [GPL License](#).

Based on the powerful editing component [Scintilla](#), Notepad++ is written in C++ and uses pure Win32 API and STL which ensures a higher execution speed and smaller program size. By optimizing as many routines as possible without losing user friendliness, Notepad++ is trying to reduce the world carbon dioxide emissions. When using less CPU power, the PC can throttle down and reduce power consumption, resulting in a greener environment.

*D:\source\notepad4ever.cpp - Notepad++

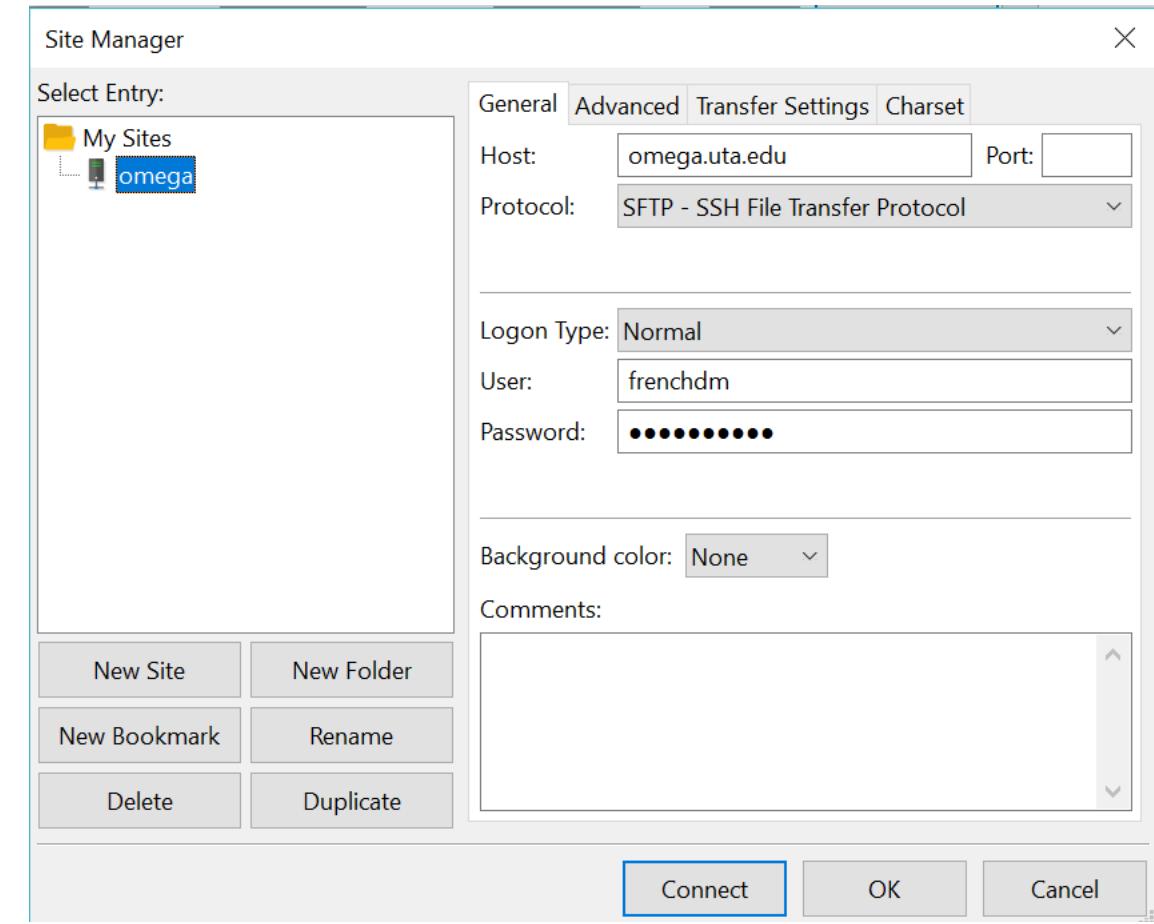
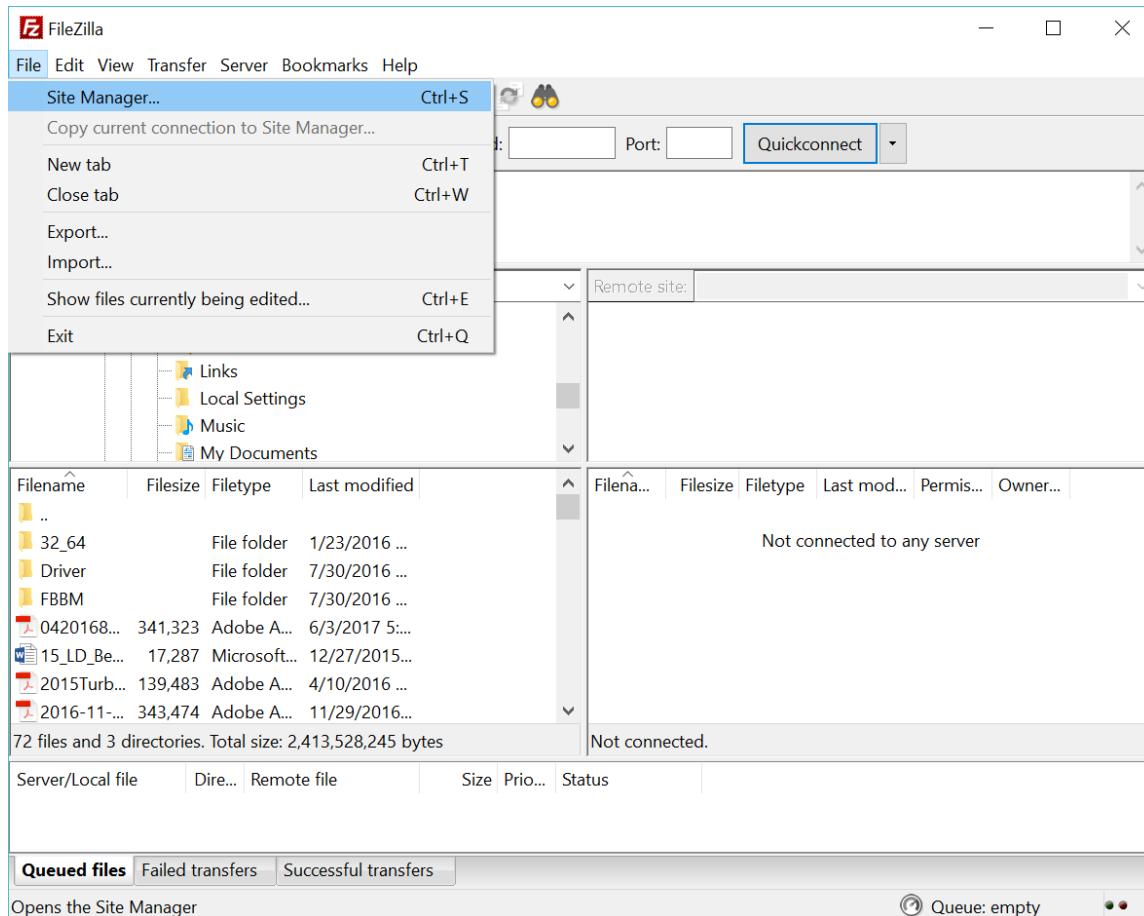
```
#include <GPL.h>
#include <free_software.h>

void notepad4ever()
{
    while (true)
    {
        Notepad++;
    }
}
```

You're encouraged to [translate Notepad++](#) into your native language if there's not already a translation present in the [Binary Translations](#) page.

I hope you enjoy Notepad++ as much as I enjoy coding it.

FileZilla



FileZilla

FileZilla

File Edit View Transfer Server Bookmarks Help

Host: omega Username: Password: Port: Quickconnect

Local site: C:\Users\Donna\Downloads\

Remote site:

| Filename | Filesize | Filetype | Last modified |
|---------------------|----------|--------------|----------------|
| .. | | | |
| 32_64 | | File folder | 1/23/2016 ... |
| Driver | | File folder | 7/30/2016 ... |
| FBBM | | File folder | 7/30/2016 ... |
| 0420168... 341,323 | 341,323 | Adobe A... | 6/3/2017 5:... |
| 15_LD_Be... 17,287 | 17,287 | Microsoft... | 12/27/2015... |
| 2015Turb... 139,483 | 139,483 | Adobe A... | 4/10/2016 ... |
| 2016-11-... 343,474 | 343,474 | Adobe A... | 11/29/2016... |

72 files and 3 directories. Total size: 2,413,528,245 bytes

Not connected.

Not connected.

Server/Local file Dire... Remote file Size Prio... Status

Queued files Failed transfers Successful transfers

Queue: empty

FileZilla - omega - sftp://frenchdm@omega.uta.edu - FileZilla

File Edit View Transfer Server Bookmarks Help

Host: omega Username: Password: Port: Quickconnect

Status: Retrieving directory listing...

Status: Listing directory /home/f/fr/frenchdm

Status: Directory listing of "/home/f/fr/frenchdm" successful

Local site: C:\Users\Donna\Downloads\

Remote site: /home/f/fr/frenchdm

| Filename | Filesize | Filetype | Last modified |
|---------------------|----------|--------------|----------------|
| .. | | | |
| 32_64 | | File folder | 1/23/2016 ... |
| Driver | | File folder | 7/30/2016 ... |
| FBBM | | File folder | 7/30/2016 ... |
| 0420168... 341,323 | 341,323 | Adobe A... | 6/3/2017 5:... |
| 15_LD_Be... 17,287 | 17,287 | Microsoft... | 12/27/2015... |
| 2015Turb... 139,483 | 139,483 | Adobe A... | 4/10/2016 ... |
| 2016-11-... 343,474 | 343,474 | Adobe A... | 11/29/2016... |

72 files and 3 directories. Total size: 2,413,528,245 bytes

| Filename | Filesize | Filetype | Last modified |
|----------|----------|-------------|------------------------|
| .. | | | |
| .mo... | | File fol... | 1/14/201... drwxr... |
| publ... | | File fol... | 1/14/201... drwxr... |
| .bas... | 31 | BASH... | 1/17/201... -rw---- |
| .bas... | 33 | BASH... | 1/14/201... -rw-r--... |
| .bas... | 141 | BASH... | 1/14/201... -rw-r--... |
| .bas... | 124 | BASHR... | 1/14/201... -rw-r--... |
| .em... | 515 | EMAC... | 1/14/201... -rw-r--... |

6 files and 2 directories. Total size: 1,502 bytes

Server/Local file Dire... Remote file Size Prio... Status

Queued files Failed transfers Successful transfers

Queue: empty

FileZilla

omega - sftp://frenchdm@omega.uta.edu - FileZilla

File Edit View Transfer Server Bookmarks Help

Host: [] Username: [] Password: [] Port: [] Quickconnect []

Status: Retrieving directory listing...

Status: Listing directory /home/f/fr/frenchdm

Status: Directory listing of "/home/f/fr/frenchdm" successful

Local site: C:\Users\Donna\Downloads

Remote site: /home/f/fr/frenchdm

| Filename | Filesize | Filetype | Last modified |
|---------------------|--------------|----------------|---------------|
| .. | | | |
| 32_64 | | File folder | 1/23/2016 ... |
| Driver | | File folder | 7/30/2016 ... |
| FBBM | | File folder | 7/30/2016 ... |
| 0420168... 341,323 | Adobe A... | 6/3/2017 5:... | |
| 15_LD_Be... 17,287 | Microsoft... | 12/27/2015... | |
| 2015Turb... 139,483 | Adobe A... | 4/10/2016 ... | |
| 2016-11... 343,474 | Adobe A... | 11/29/2016... | |

72 files and 3 directories. Total size: 2,413,528,245 bytes

| Filename | Filesize | Filetype | Last mod... | Permis... | Owner... |
|----------|----------|-------------|-------------|-----------|-----------|
| .. | | | | | |
| .mo... | | File fol... | 1/14/201... | drwxr- | french... |
| publ... | | File fol... | 1/14/201... | drwxr- | french... |
| .bas... | 31 | BASH... | 1/17/201... | -rw---- | french... |
| .bas... | 33 | BASH... | 1/14/201... | -rw-r-- | french... |
| .bas... | 141 | BASH... | 1/14/201... | -rw-r-- | french... |
| .bas... | 124 | BASHR... | 1/14/201... | -rw-r-- | french... |
| .em... | 515 | EMAC... | 1/14/201... | -rw-r-- | french... |

6 files and 2 directories. Total size: 1,502 bytes

Server/Local file Dire... Remote file Size Prio... Status

Queued files Failed transfers Successful transfers

Queue: empty

omega - sftp://frenchdm@omega.uta.edu - FileZilla

File Edit View Transfer Server Bookmarks Help

Host: [] Username: [] Password: [] Port: [] Quickconnect []

Status: Retrieving directory listing of "/home/f/fr/frenchdm"...

Status: Listing directory /home/f/fr/frenchdm

Status: Directory listing of "/home/f/fr/frenchdm" successful

Local site: C:\Users\Donna\Desktop\UTA\Class Periods\Week1.Day2\

Remote site: /home/f/fr/frenchdm

| Filename | Filesize | Filetype | Last mod... | Permis... | Owner... |
|-------------------|----------|-------------|-------------|-----------|----------|
| .. | | | | | |
| OmegaTestFile.txt | 0 | Text Doc... | 1/18/201... | | |

Selected 1 file. Total size: 0 bytes

| Filename | Filesize | Filetype | Last mod... | Permis... | Owner... |
|-------------------|----------|-----------|-------------|-----------|----------|
| .bash_history | 31 | BASH... | 1/17/201... | -rw---- | |
| .bash_logout | 33 | BASH... | 1/14/201... | -rw-r-- | |
| .bash_profile | 141 | BASH... | 1/14/201... | -rw-r-- | |
| .bashrc | 124 | BASHR... | 1/14/201... | -rw-r-- | |
| .emacs | 515 | EMAC... | 1/14/201... | -rw-r-- | |
| .zshrc | 658 | ZSHRC... | 1/14/201... | -rw-r-- | |
| OmegaTestFile.txt | 0 | Text D... | 1/18/201... | -rw-r-- | |

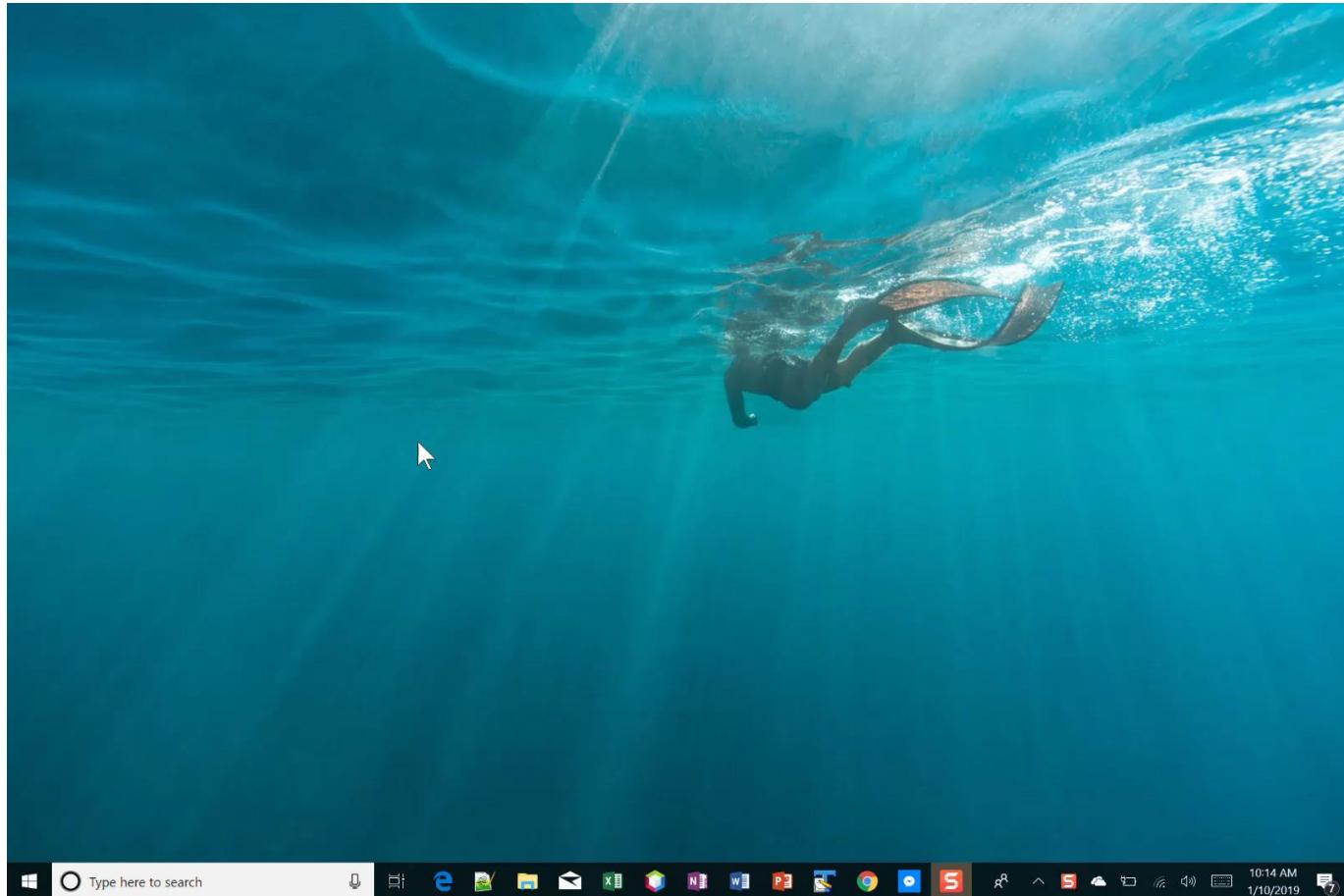
7 files and 2 directories. Total size: 1,502 bytes

Server/Local file Dire... Remote file Size Prio... Status

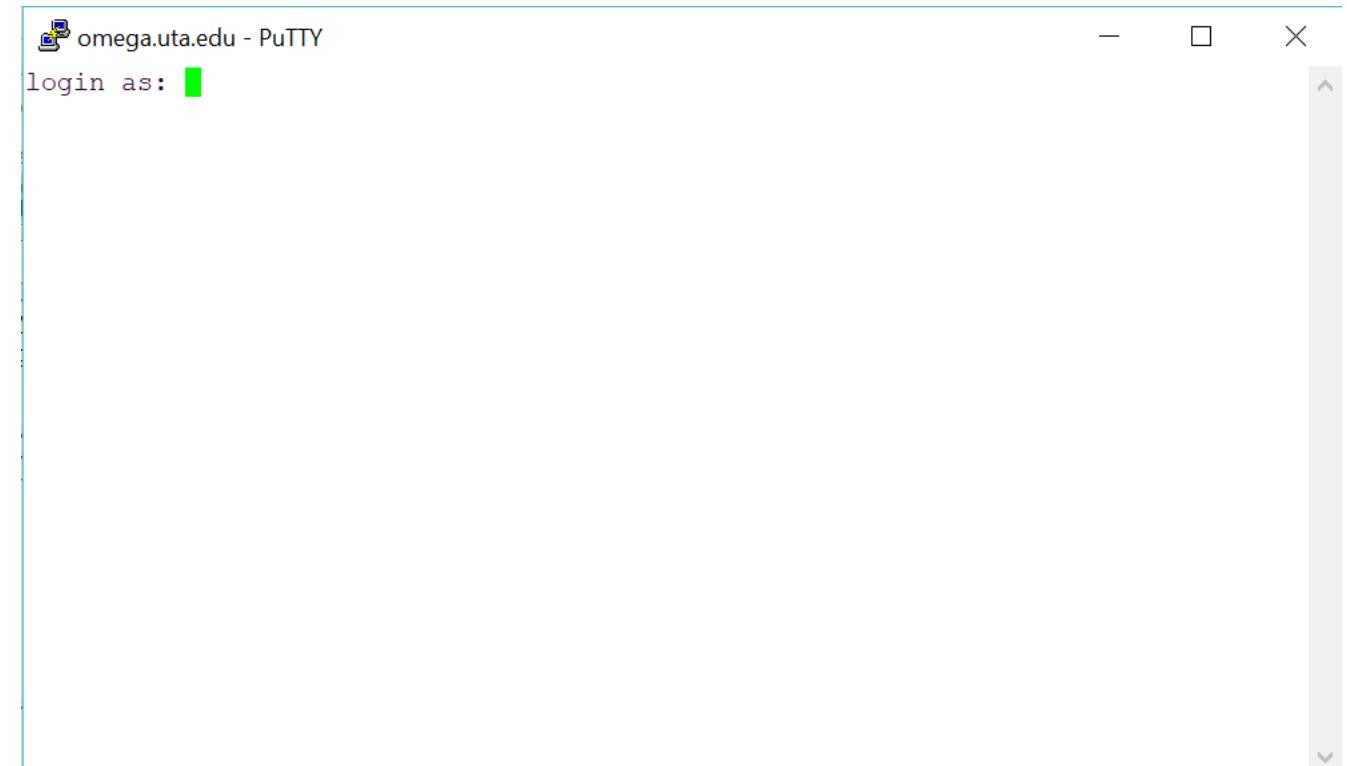
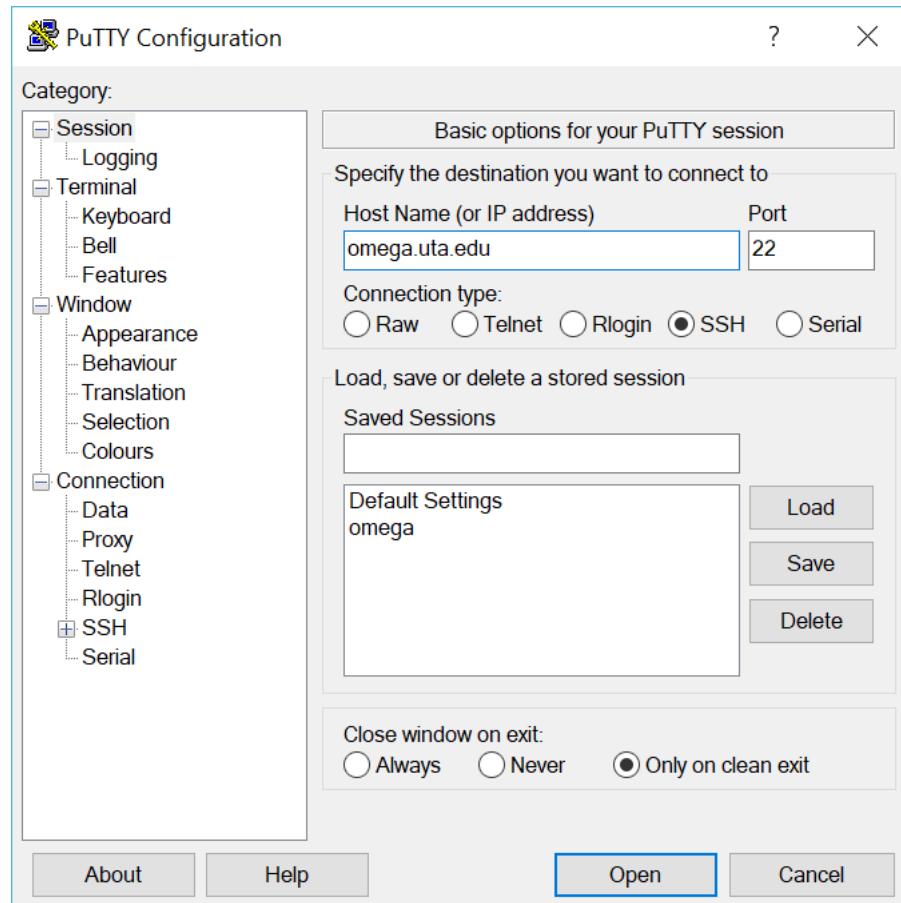
Queued files Failed transfers Successful transfers (1)

Queue: empty

FileZilla



PuTTY



PuTTY

omega.uta.edu - PuTTY

login as: frenchdm

This UT Arlington information resource, including all related equipment, networks and network devices, is provided for authorized use only. All unauthorized use of this information resource is prohibited. Misuse is subject to criminal prosecution and/or administrative or other disciplinary action.

Usage of this information resource, authorized or unauthorized, may be subject to security testing and monitoring. In addition, all information, including personal information that is placed on or sent over this resource is the property of the State of Texas and may also be subject to security testing and monitoring. Evidence of unauthorized use and/or misuse collected during security testing and monitoring is subject to criminal prosecution and/or administrative or other disciplinary action.

Usage of this information resource constitutes consent to all policies and procedures set forth by UT Arlington and there is no expectation of privacy except as otherwise provided by applicable privacy laws.

frenchdm@omega.uta.edu's password: [REDACTED]

frenchdm@omega:~

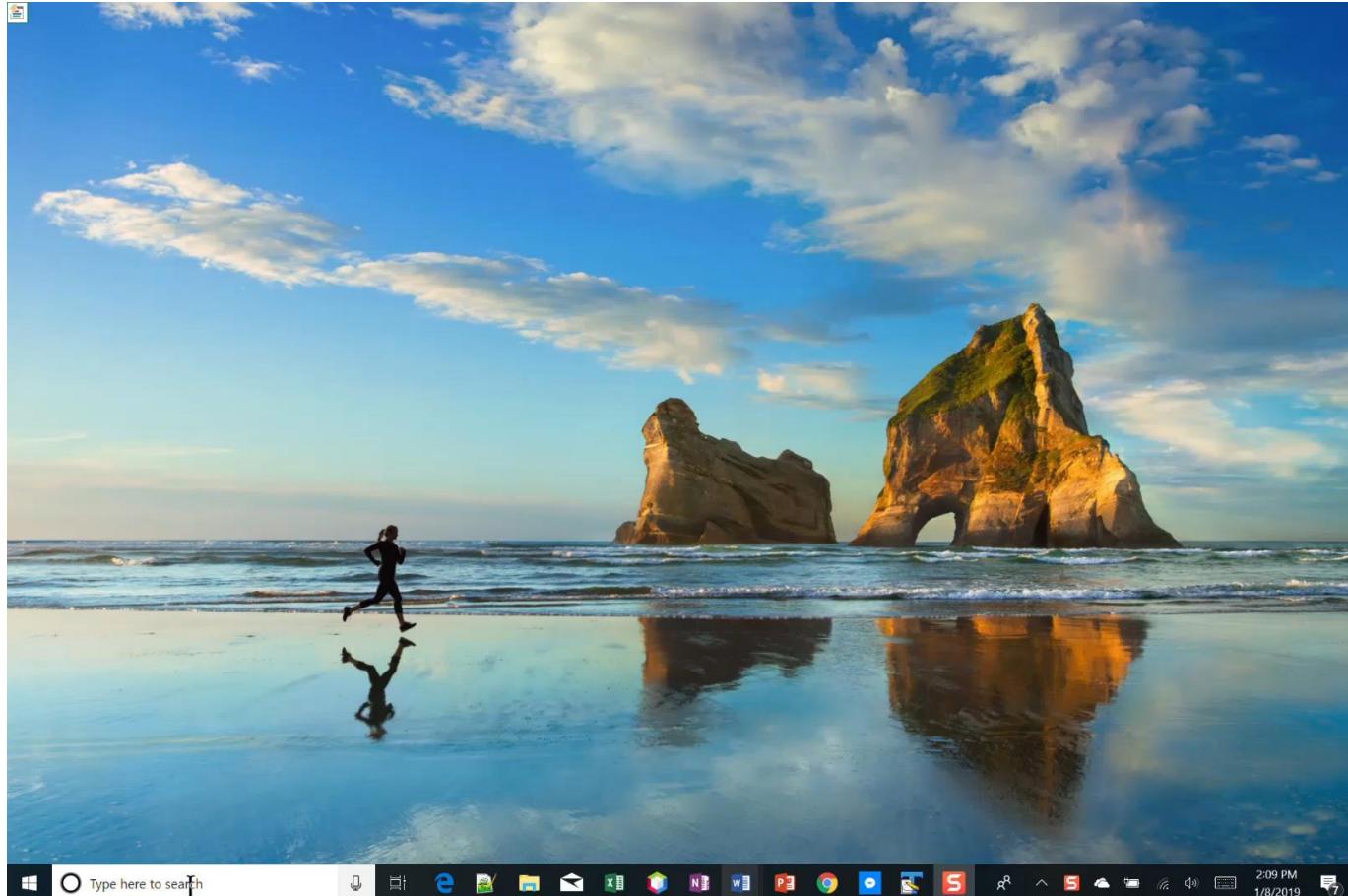
This UT Arlington information resource, including all related equipment, networks and network devices, is provided for authorized use only. All unauthorized use of this information resource is prohibited. Misuse is subject to criminal prosecution and/or administrative or other disciplinary action.

Usage of this information resource, authorized or unauthorized, may be subject to security testing and monitoring. In addition, all information, including personal information that is placed on or sent over this resource is the property of the State of Texas and may also be subject to security testing and monitoring. Evidence of unauthorized use and/or misuse collected during security testing and monitoring is subject to criminal prosecution and/or administrative or other disciplinary action.

Usage of this information resource constitutes consent to all policies and procedures set forth by UT Arlington and there is no expectation of privacy except as otherwise provided by applicable privacy laws.

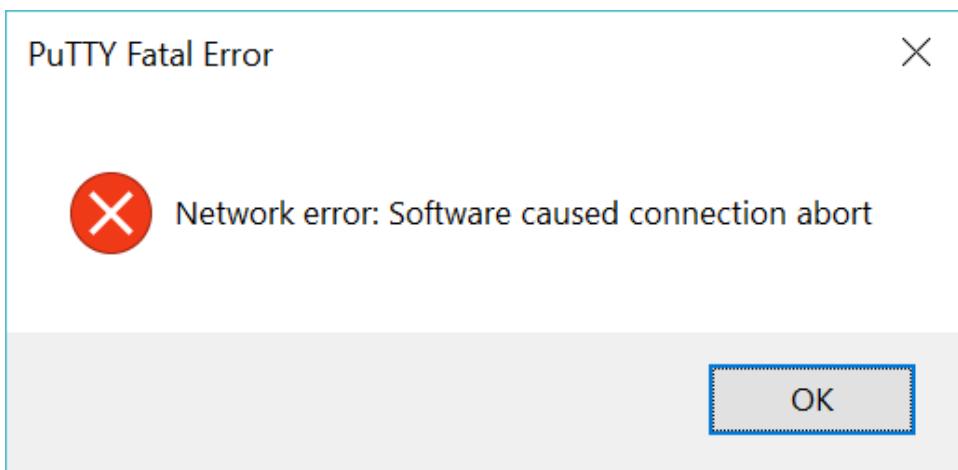
frenchdm@omega.uta.edu's password:
Last login: Thu Jan 18 18:39:09 2018 from 71-91-162-160.dhcp.gwnt.ga.charter.com
[frenchdm@omega ~]\$ [REDACTED]

PuTTY

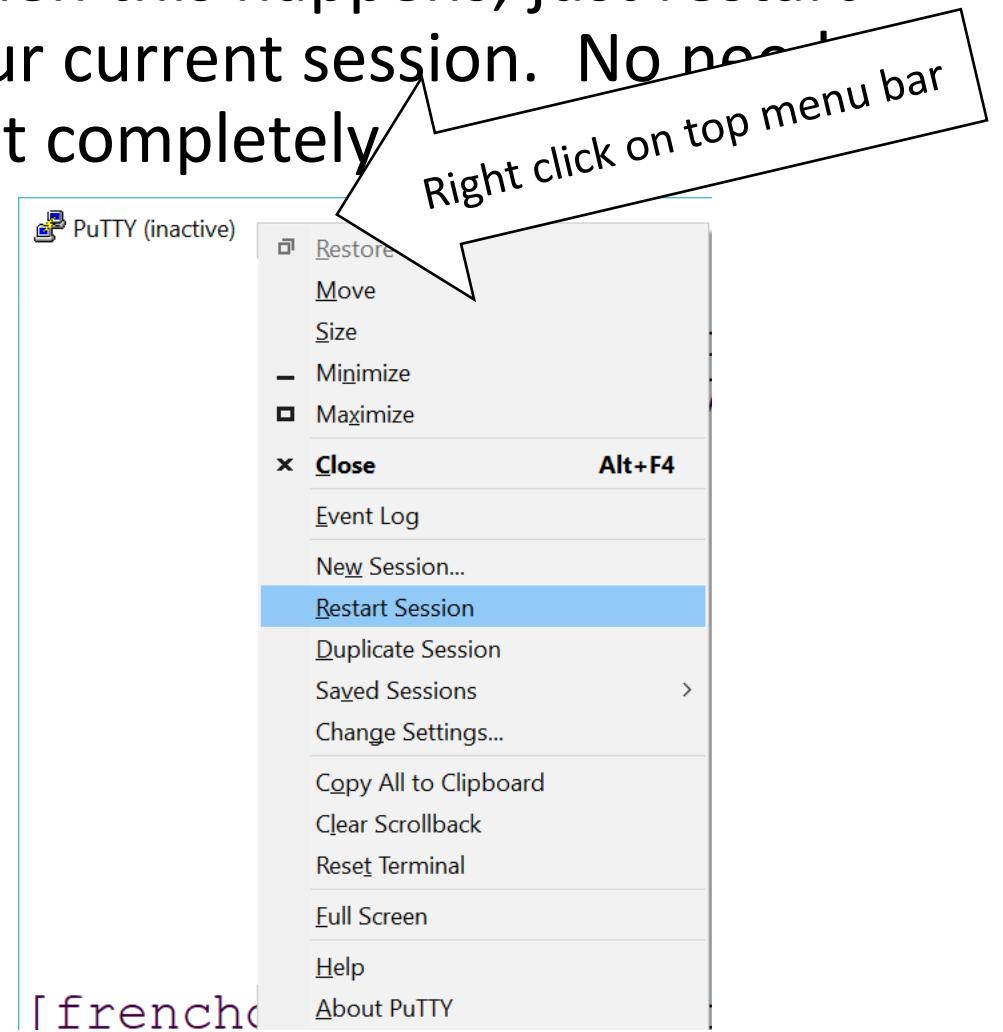


PuTTY

If you leave an Omega session idle for too long, it will automatically disconnect.

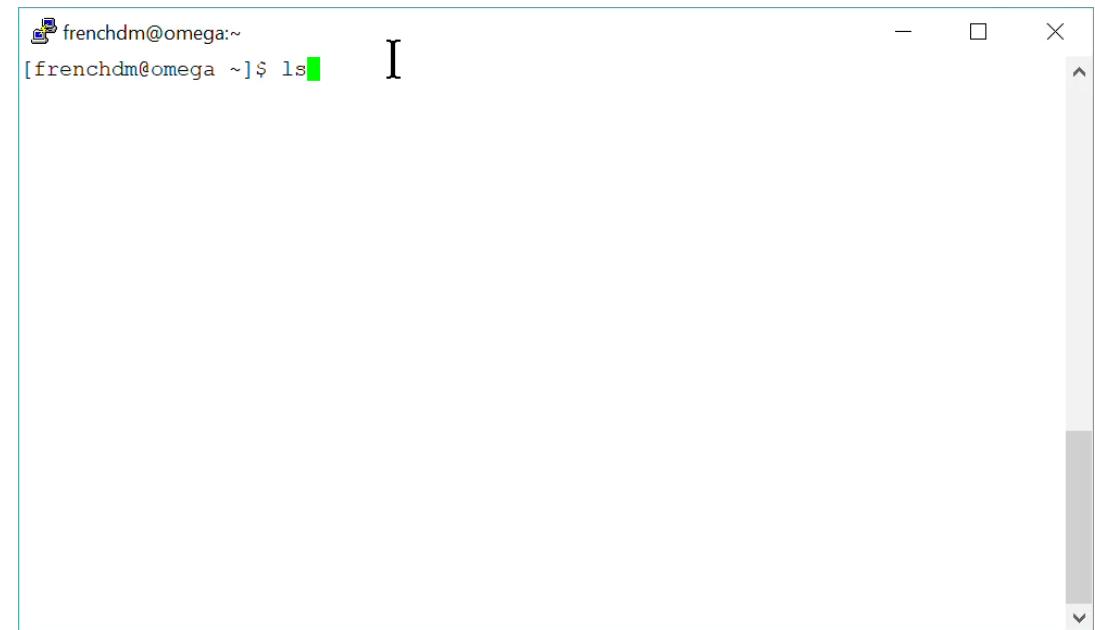


When this happens, just restart your current session. No need to quit completely

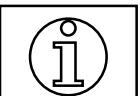


UNIX and Omega

- When you first login, you will be in your home directory which will have the same name your user name, for example, **frenchdm**, and it is where your personal files and subdirectories are saved.
- The **ls** command lists the contents of your current working directory.
- The **ls -a** command lists files that are normally hidden.
- The **ls -l** command lists more information about the files

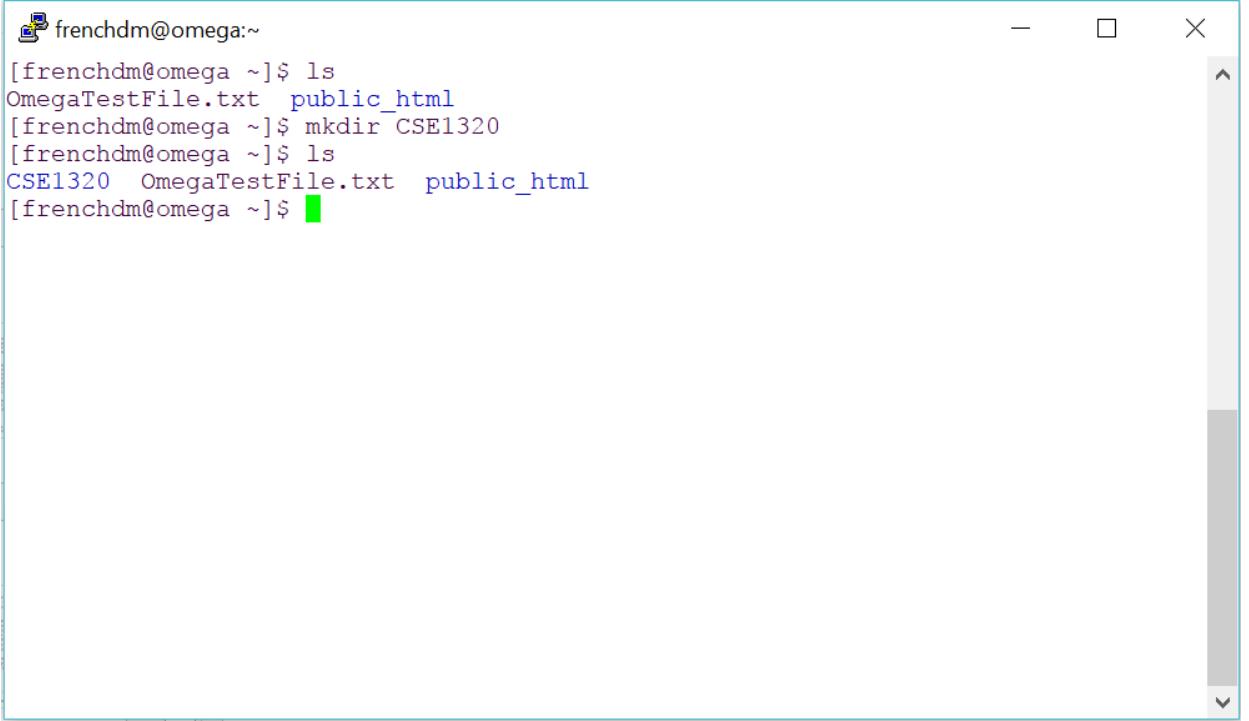


A screenshot of a terminal window titled "frenchdm@omega:~". The window shows the command "[frenchdm@omega ~]\$ ls" entered by the user. The terminal is currently empty, displaying only the command prompt and the command itself.



UNIX and Omega

- The `mkdir` command is used to **make subdirectories** in your home directory. Let's make a subdirectory called "CSE1320".
- **`mkdir CSE1320`**



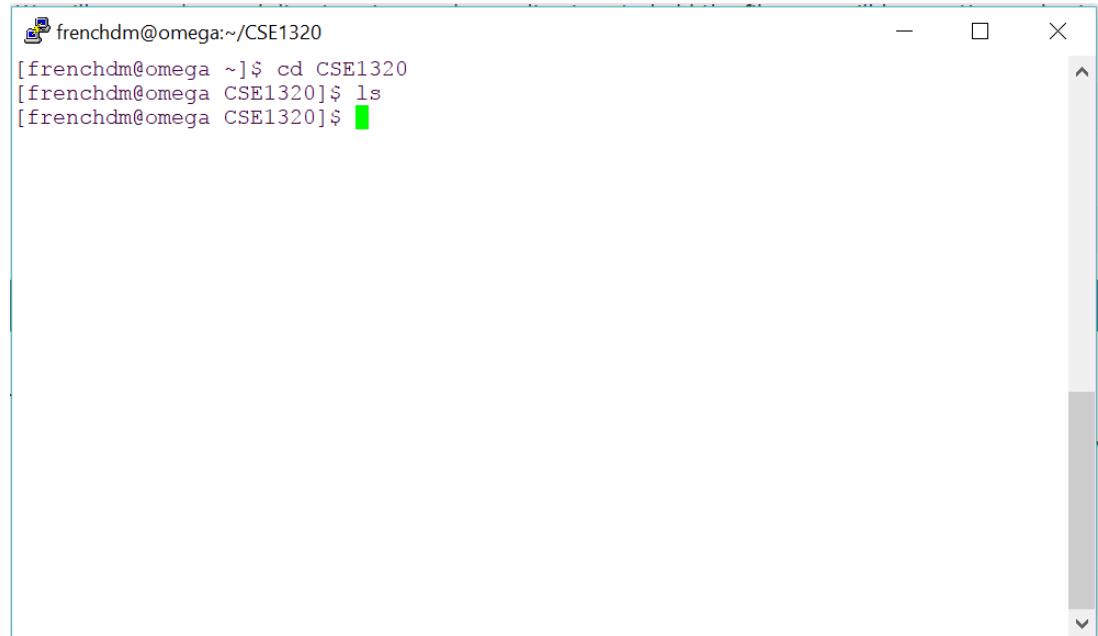
A screenshot of a terminal window titled "frenchdm@omega:~". The window shows a command-line session:

```
[frenchdm@omega ~]$ ls  
OmegaTestFile.txt public_html  
[frenchdm@omega ~]$ mkdir CSE1320  
[frenchdm@omega ~]$ ls  
CSE1320 OmegaTestFile.txt public_html  
[frenchdm@omega ~]$
```

The terminal has a light blue background and a white foreground. The text is black, except for the file names which are colored blue. A small green square cursor is visible at the end of the last command line.

UNIX and Omega

- The **change directory** command allows you to change from the current working directory to a different directory.
- **cd CSE1320**



```
frenchdm@omega:~/CSE1320
[frenchdm@omega ~]$ cd CSE1320
[frenchdm@omega CSE1320]$ ls
[frenchdm@omega CSE1320]$ █
```

A screenshot of a terminal window titled "frenchdm@omega:~/CSE1320". The window contains three lines of text: "[frenchdm@omega ~]\$ cd CSE1320", "[frenchdm@omega CSE1320]\$ ls", and "[frenchdm@omega CSE1320]\$ █". The terminal has a light blue background and a white foreground. It features standard window controls (minimize, maximize, close) at the top right and scroll bars on the right side.

UNIX and Omega

cd

- Returns you to your home directory

cd .

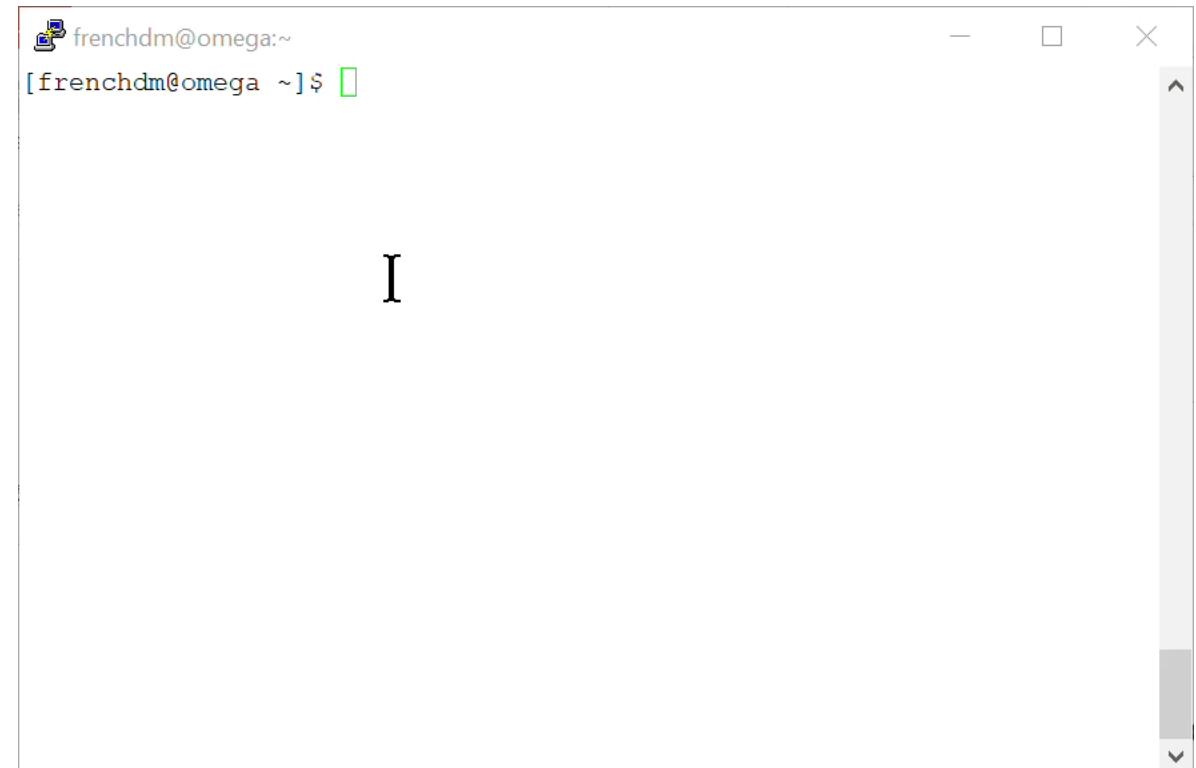
- Keeps you in the current directory

cd ..

- Takes you one directory back

cd –

- Returns you to whatever directory you were in last

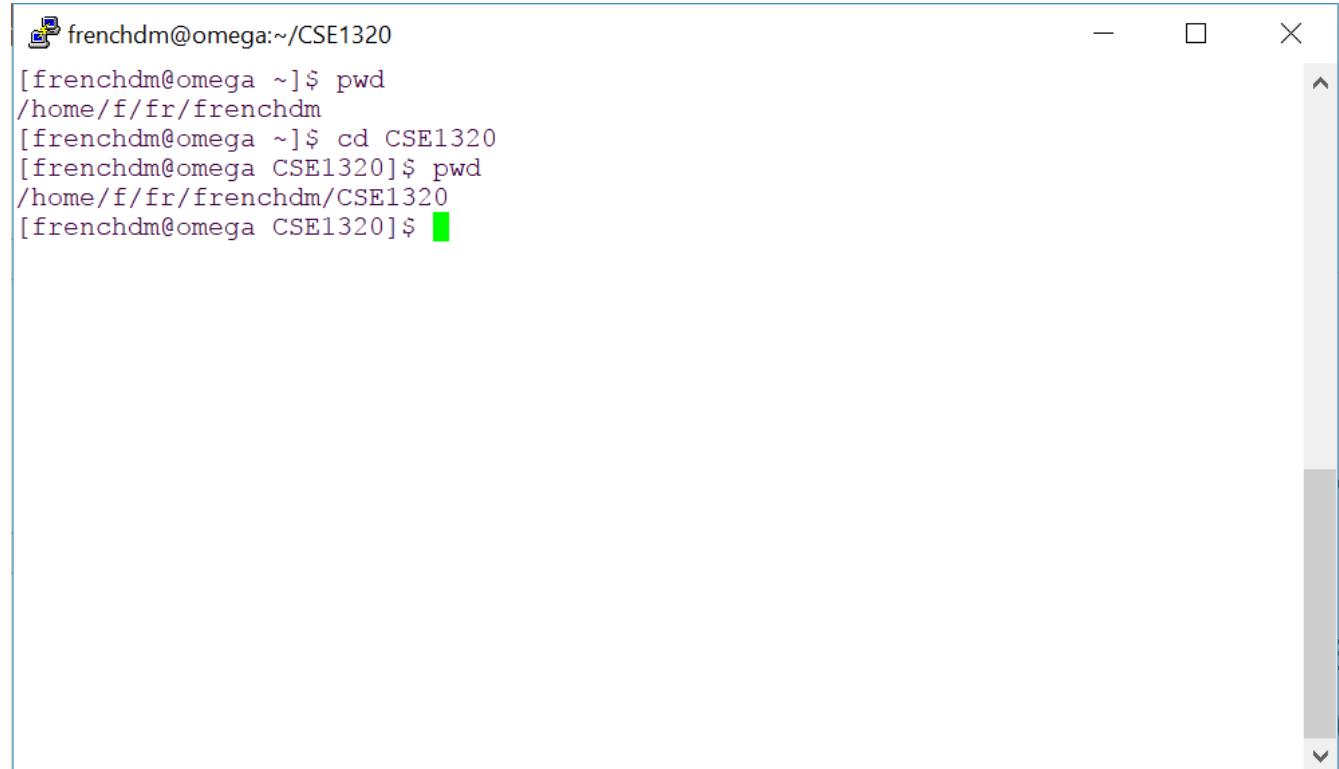


A screenshot of a terminal window titled "frenchdm@omega:~". The window shows a command prompt "[frenchdm@omega ~]\$". A cursor is visible in the text area.



UNIX and Omega

The **print working directory** command, **pwd**, shows you where you are.



A screenshot of a terminal window titled "frenchdm@omega:~/CSE1320". The window contains the following text:

```
[frenchdm@omega ~]$ pwd  
/home/f/fr/frenchdm  
[frenchdm@omega ~]$ cd CSE1320  
[frenchdm@omega CSE1320]$ pwd  
/home/f/fr/frenchdm/CSE1320  
[frenchdm@omega CSE1320]$
```

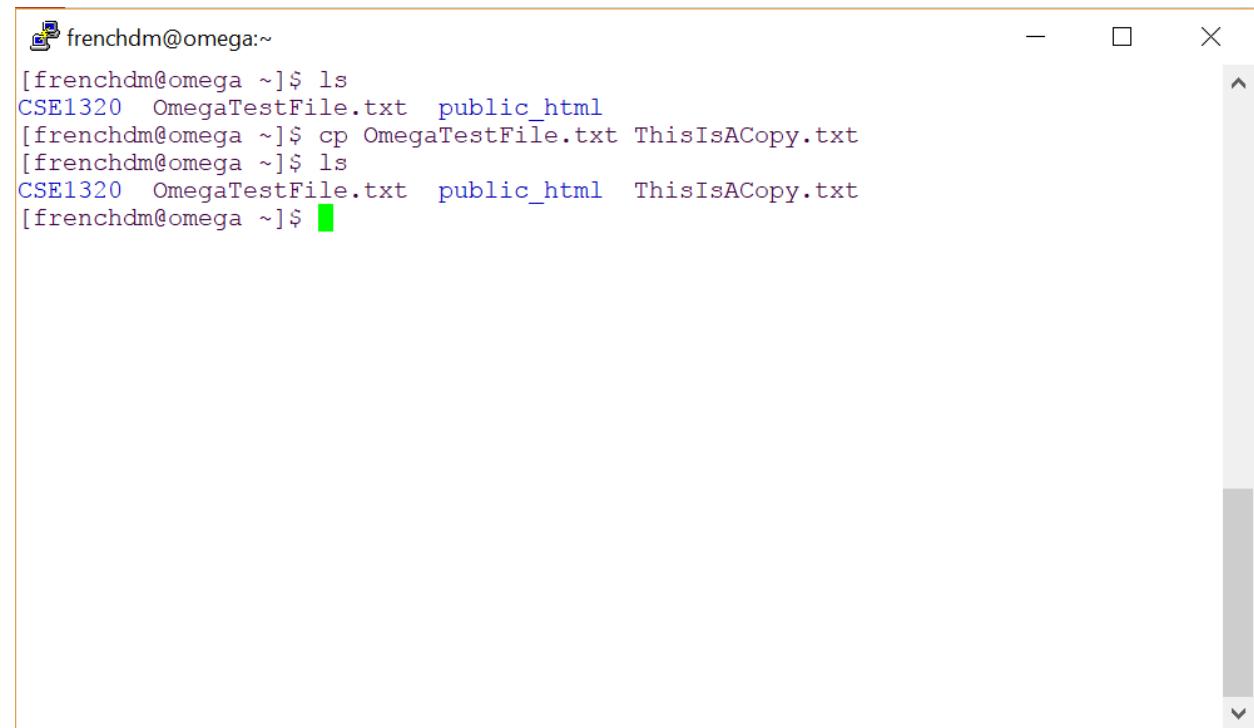
The terminal window has standard operating system window controls (minimize, maximize, close) in the top right corner. A vertical scroll bar is visible on the right side of the window.

UNIX and Omega

cp file1 file2

is the command which makes a
copy of *file1* in the current
working directory and calls it *file2*

I copied OmegaTestFile.txt to
ThisIsACopy.txt



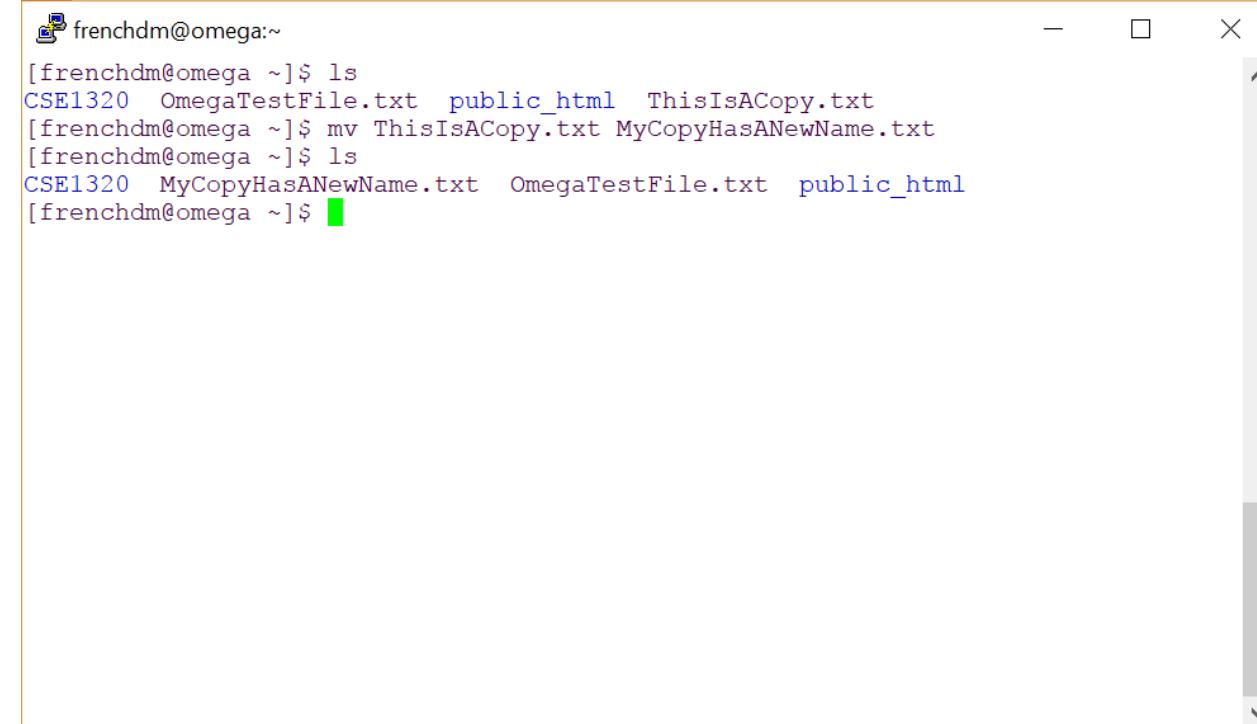
```
frenchdm@omega:~  
[frenchdm@omega ~]$ ls  
CSE1320 OmegaTestFile.txt public_html  
[frenchdm@omega ~]$ cp OmegaTestFile.txt ThisIsACopy.txt  
[frenchdm@omega ~]$ ls  
CSE1320 OmegaTestFile.txt public_html ThisIsACopy.txt  
[frenchdm@omega ~]$
```

UNIX and Omega

mv file1 file2

is the command which **moves** or renames a file

I renamed ThisIsACopy.txt to MyCopyHasANewName.txt



```
frenchdm@omega:~$ ls  
CSE1320 OmegaTestFile.txt public_html ThisIsACopy.txt  
[frenchdm@omega ~]$ mv ThisIsACopy.txt MyCopyHasANewName.txt  
[frenchdm@omega ~]$ ls  
CSE1320 MyCopyHasANewName.txt OmegaTestFile.txt public_html  
[frenchdm@omega ~]$ █
```

A screenshot of a terminal window titled 'frenchdm@omega:~'. The window shows a command-line session. The user first lists files with 'ls', showing 'CSE1320', 'OmegaTestFile.txt', 'public_html', and 'ThisIsACopy.txt'. Then, the user runs the command 'mv ThisIsACopy.txt MyCopyHasANewName.txt'. Finally, the user lists files again with 'ls', showing 'CSE1320', 'MyCopyHasANewName.txt', 'OmegaTestFile.txt', and 'public_html'. A green cursor is visible at the end of the command line.

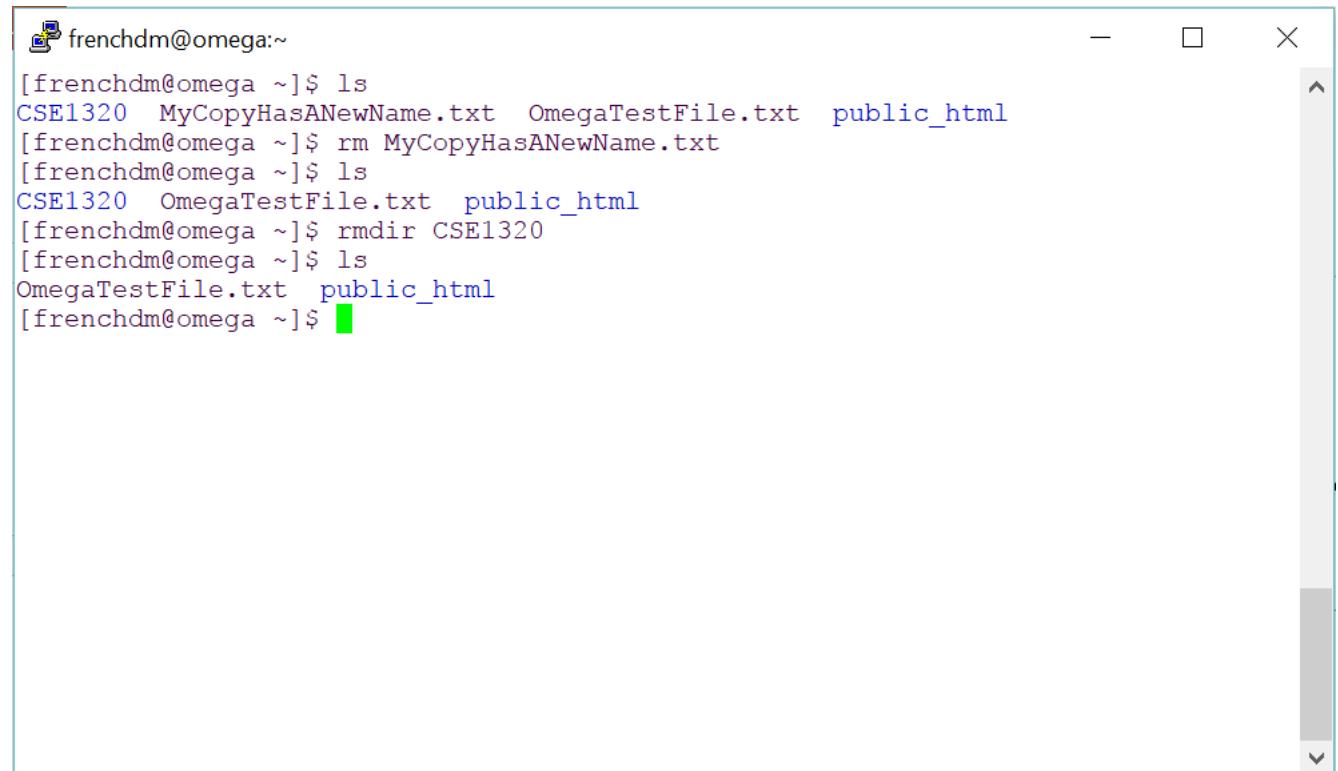
UNIX and Omega

rm file

removes/deletes a file

rmdir directory

removes/deletes a directory

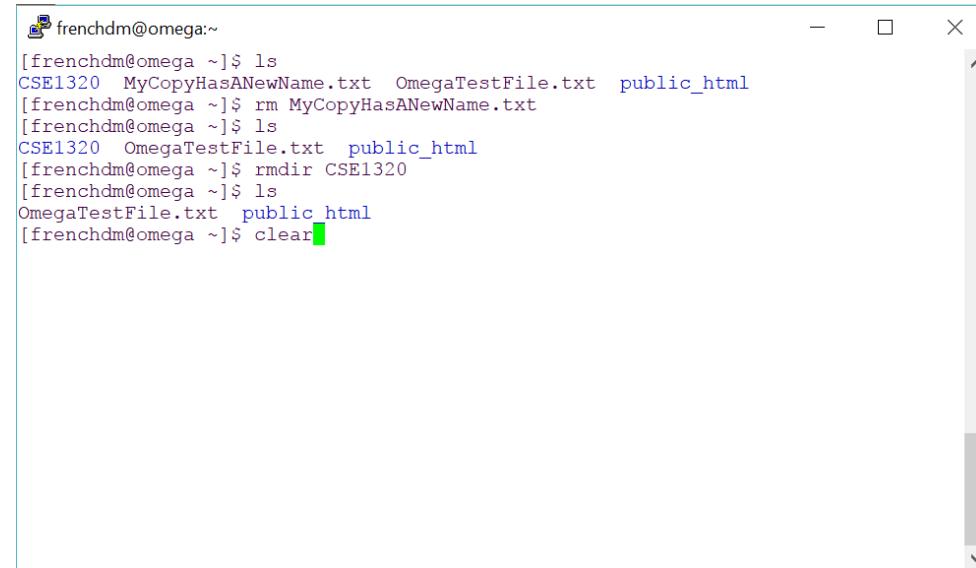


```
frenchdm@omega:~  
[frenchdm@omega ~]$ ls  
CSE1320 MyCopyHasANewName.txt OmegaTestFile.txt public_html  
[frenchdm@omega ~]$ rm MyCopyHasANewName.txt  
[frenchdm@omega ~]$ ls  
CSE1320 OmegaTestFile.txt public_html  
[frenchdm@omega ~]$ rmdir CSE1320  
[frenchdm@omega ~]$ ls  
OmegaTestFile.txt public_html  
[frenchdm@omega ~]$
```

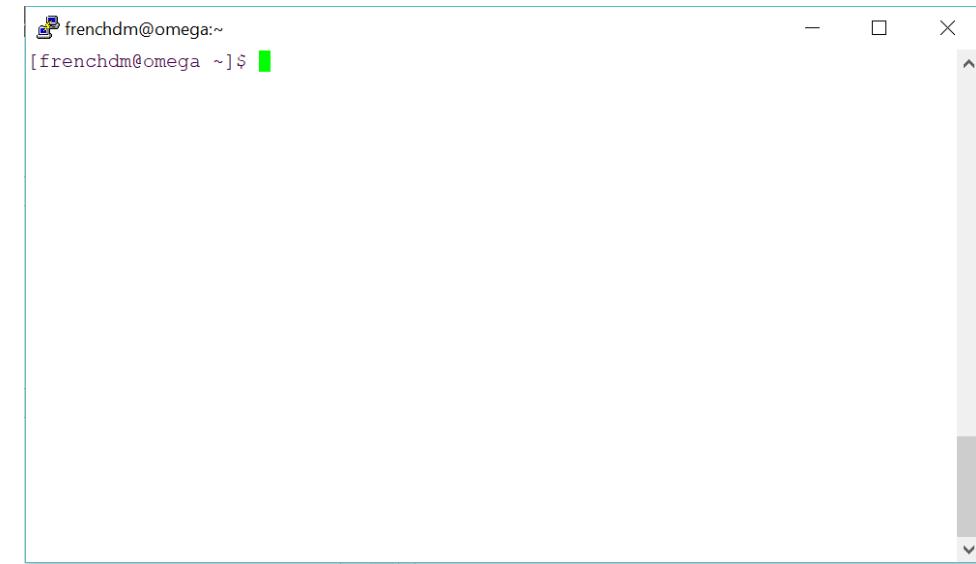
UNIX and Omega

clear

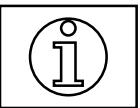
This will clear all text and leave you with the prompt at the top of the window.



frenchdm@omega:~
[frenchdm@omega ~]\$ ls
CSE1320 MyCopyHasANewName.txt OmegaTestFile.txt public_html
[frenchdm@omega ~]\$ rm MyCopyHasANewName.txt
[frenchdm@omega ~]\$ ls
CSE1320 OmegaTestFile.txt public_html
[frenchdm@omega ~]\$ rmdir CSE1320
[frenchdm@omega ~]\$ ls
OmegaTestFile.txt public_html
[frenchdm@omega ~]\$ clear



frenchdm@omega:~
[frenchdm@omega ~]\$



UNIX and Omega

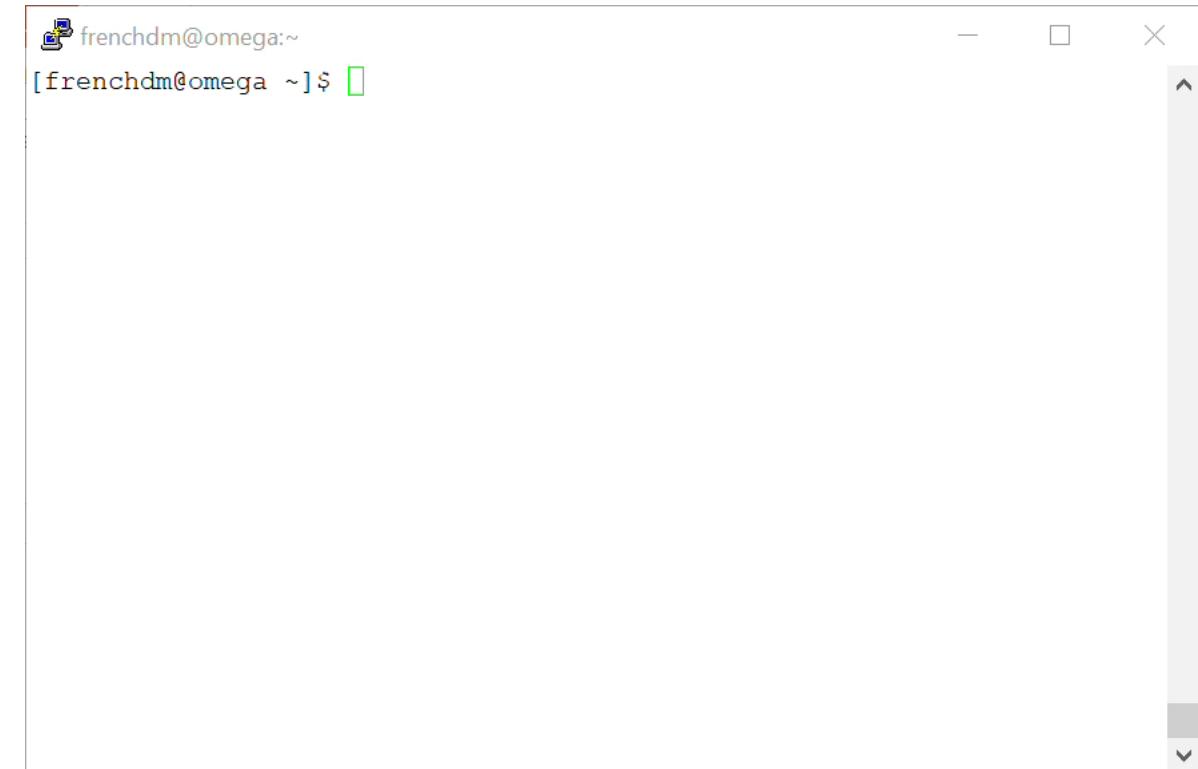
cat *filename*

The command **cat** can be used to display the contents of a file on the screen.

more *filename*

The command **more** can be used to display the contents of a file on the screen.

Press the **SPACEBAR** if you want to see another page, press **ENTER** to see the next line and type **q** if you want to quit reading.



UNIX and Omega

apropos xxx

The command **apropos** can
be used to display any
command that references
xxx.

ap·ro·pos

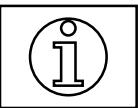
/ˌaprəˈpō/ 🔍

preposition

1. with reference to; concerning.



A screenshot of a terminal window titled 'frenchdm@omega:~'. The window shows the command '[frenchdm@omega ~]\$ apropos ~' being typed. The terminal has standard window controls (minimize, maximize, close) and a vertical scroll bar on the right side.

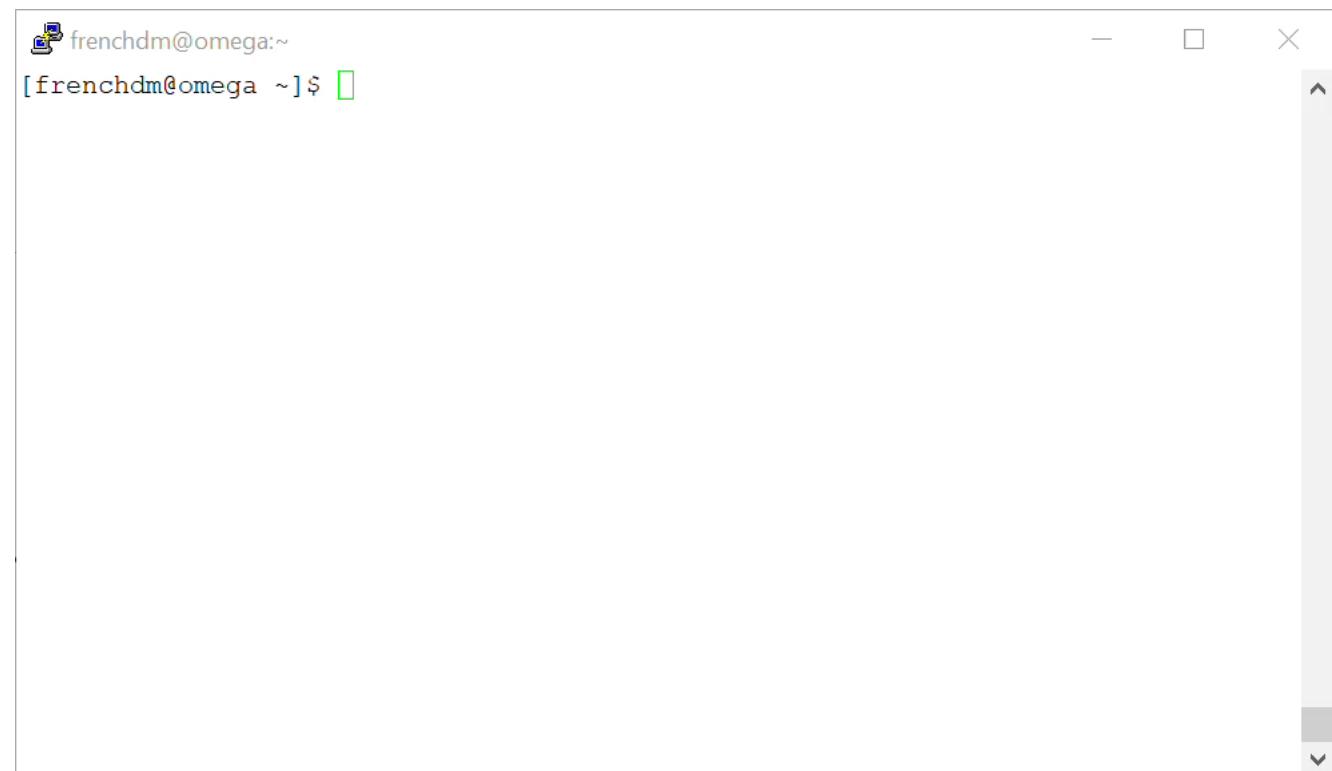


UNIX and Omega

man xxx

The command **man** can
be used to display the manual
built into UNIX for that
command

- use **q** to quit/exit man
- use **ENTER** to scroll line by line
- use **SPACEBAR** to scroll page by page

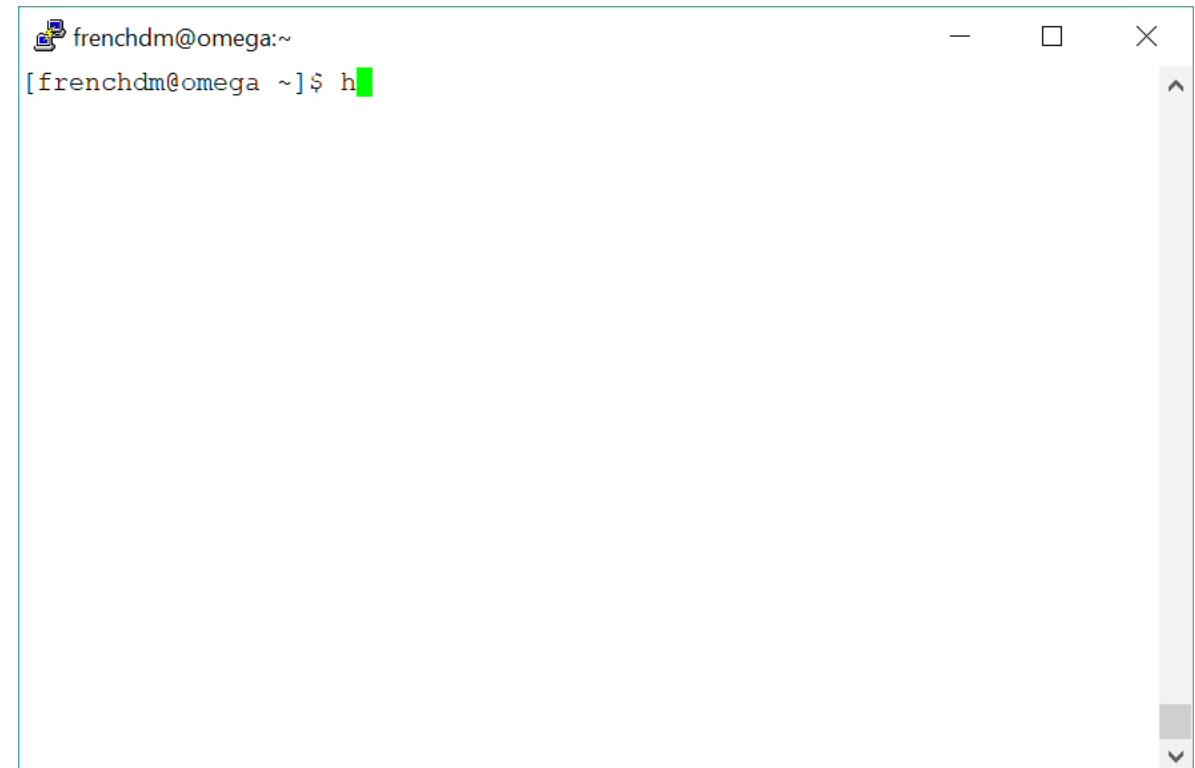


A screenshot of a terminal window titled "frenchdm@omega:~". The window shows a command prompt "[frenchdm@omega ~]\$". The main area of the terminal is completely blank, indicating that no output has been displayed yet.

UNIX and Omega

Keyboard Shortcuts

- The arrow keys
- TAB completion
- Exclamation Point
- Ctrl-r (reverse search)
- history
- history | grep
- Ctrl-t



A screenshot of a terminal window titled "frenchdm@omega:~". The window shows the command line prompt "[frenchdm@omega ~]\$ h" followed by a green cursor. The window has standard operating system window controls (minimize, maximize, close) and a vertical scroll bar on the right.



- 2192-CSE-1320-002-INTERMEDIATE-PROGRAMMING--2019-Spring
- Announcements
- Syllabus
- Course Materials
- Calendar
- GTA Contact Info
- Crash Course
- Homework
- Coding Assignments
- UTA Email
- Grades
- Student Help and Resources
- Groups
- Discussions
- Degree Map
- Inspire for Faculty

Course Management

- Control Panel
 - File Directory
 - Course Tools
 - Evaluation
- Grade Center
 - Needs Grading
 - Full Grade Center
- Users and Groups
- Customization

UNIX Commands

If this item does not open automatically you can [open UNIX Commands here](#)

| Command | Meaning |
|------------------------|--|
| ls | list files and directories |
| ls -a | list all files and directories |
| ls -l | list all files and information about those files |
| mkdir <i>directory</i> | make a directory |
| cd <i>directory</i> | change to named directory |
| cd | change to home directory |
| cd . | stay in current directory |
| cd .. | change to parent directory |
| cd - | returns you to whatever direct you were in last |
| pwd | display the path of the current directory |
| cp <i>file1 file2</i> | copy <i>file1</i> and call it <i>file2</i> |
| mv <i>file1 file2</i> | move or rename <i>file1</i> to <i>file2</i> |
| rm <i>file1</i> | remove/delete <i>file1</i> |
| rmdir <i>directory</i> | remove a directory |
| clear | clear the screen |

Your First C Program

Use your choice of text editor to write the code. I am using Notepad++. Be sure to add a blank line at the end.

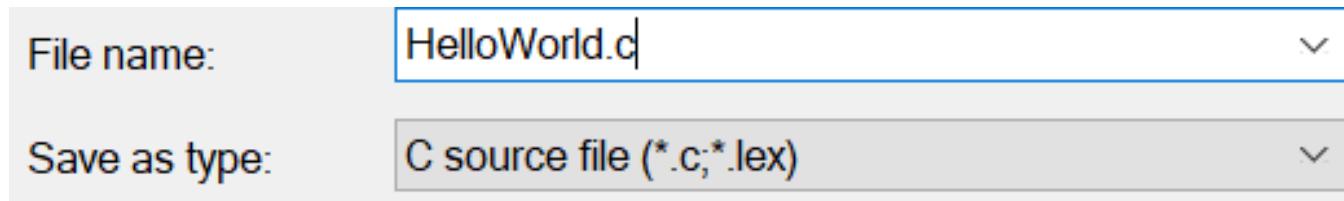


```
1 /* Donna French 1000074079 */
2
3 /* This is my first C program for CSE 1320 */
4
5 #include <stdio.h>
6
7 int main(void)
8 {
9     printf("Hello World\n");
10    return 0;
11 }
12
```



Your First C Program

Notepad++ can be told to save your file as a .c file.

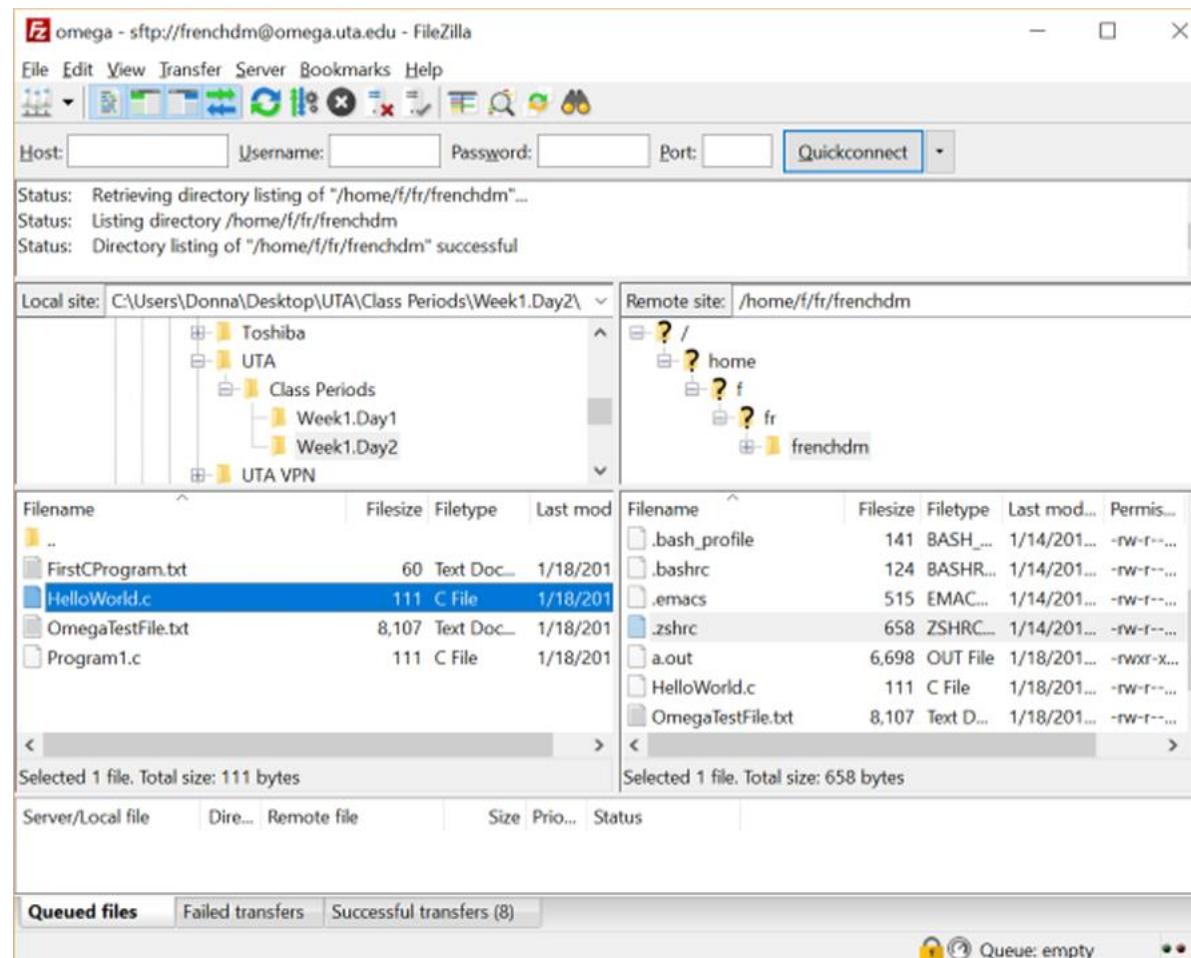


Saving it as a .c file enables the C specific colored syntax.



Your First C Program

FTP the file to Omega



Your First C Program

- Compile your program on Omega

```
gcc HelloWorld.c
```

- Run your program

```
a.out
```

- See your output

```
Hello World
```

```
frenchdm@omega:~  
[frenchdm@omega ~]$ ls  
HelloWorld.c OmegaTestFile.txt Program1.c public_html  
[frenchdm@omega ~]$ gcc HelloWorld.c  
[frenchdm@omega ~]$ ls  
a.out HelloWorld.c OmegaTestFile.txt Program1.c public_html  
[frenchdm@omega ~]$ a.out  
Hello World  
[frenchdm@omega ~]$ █
```



Type here to search



12:31 PM

1/10/2019



Language Level

Computer languages can be

high level

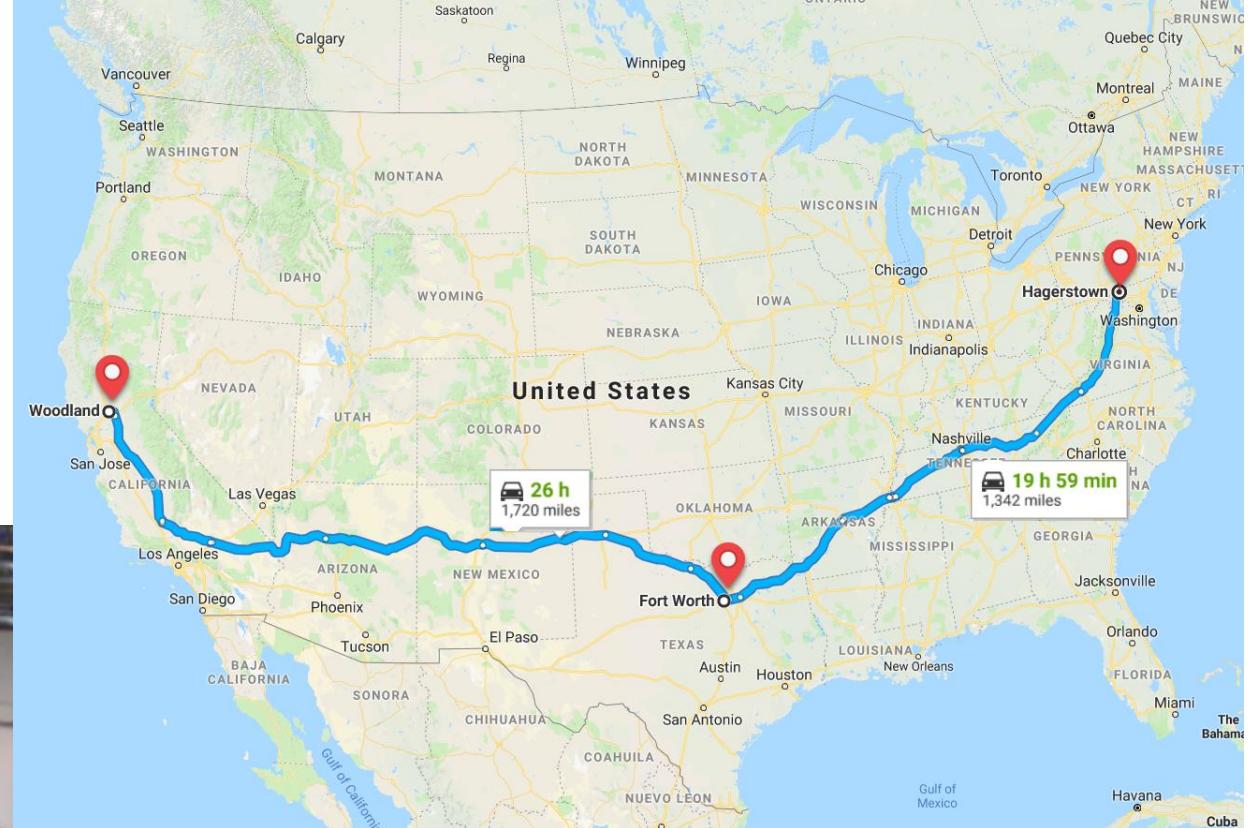
intermediate level

low level

Low level – assembly language – used to write operating systems

Higher level – makes programs easier to port between systems

C is an intermediate to high level language because it allows programmers to have some control over the hardware. Assembly code can be written into a C program.



Reserved Words

Reserved words or keywords in C are the words that are part of the language itself. There is a complete list in the Programmer's Handbook.

```
File Edit Search View Encoding Languages Tools Macro Run Plugins  
Program1.c  
1 /* This is a comment  
2  
3 #include <stdio.h>  
4  
5 int main()  
6 {  
7     printf("Hello World\n");  
8     return 0;  
9 }  
10
```

#include is a preprocessor command

stdio.h is a library

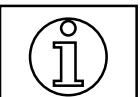
printf() is a function

{ } are tokens



Identifiers or Names in C

- Can consist of letters, digits and underscores
- Must start with a letter or underscore.
- C is case sensitive.
- Must not be identical to a keyword
- Length is limited – varies between compilers



The Character Set

- American Standard Code for Information Interchange - ASCII
 - All the letters are in consecutive order. Capital letters are grouped together while small letters also have their own group.
- Extended Binary Coded Decimal Interchange Code - EBCDIC
 - The letters are grouped 9 at a time. Stems from punch card origins and is quite difficult for programmers to deal with.

To learn more

The Science Elf – Format Wars : ASCII vs EBCDIC

<https://www.youtube.com/watch?v=3kXLHLUhV5Q>

Course Materials -> Videos From Class



Format of C Programs

C is a free-format language.

- No requirements that code begin in a certain column
- No requirements that statements must be contained on a single line
- No requirements that comments must be located in a special place

White space

- Space
- Line feed
- Backspace
- Horizontal tab
- Vertical tab
- Form feed
- Carriage return



Format of C Programs

```
/* Donna French 1000074079 */  
  
/* This is my first C program for CSE 1320 */  
  
#include <stdio.h>  
  
int main(void)  
{  
    printf("Hello World");  
    return 0;  
}
```

```
#include <stdio.h>  
int main(void) {printf("Hello  
World");return 0; }
```

Format of C Programs

- Student Name and ID will be in comment(s) at the start of every program
- All indentions will use at least 3 spaces and a maximum of 5 spaces. The tab character in Notepad++ defaults to 4 spaces.
- Formatting of code will count as 10% of the final grade of each coding assignment. We will discuss the required standard as different parts of the language are introduced.
- The goal is to form good habits that will help you going forward in your Computer Science academic career and future professional career.



main()

- every C program must have a function named main ()
- main () must return an int
- placement of main () within a program
 - at the bottom
 - below everything else
 - not at the bottom
 - requires the use of prototypes

The screenshot shows a C code editor interface with a menu bar and a toolbar. The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, and Plugins. The toolbar has various icons for file operations like Open, Save, Print, and Build. A window titled "Program1.c" is open, displaying the following C code:

```
1  /* This is a comment */
2
3  #include <stdio.h>
4
5  int main (void)
6  {
7      printf("Hello World\n");
8      return 0;
9
10 }
```



Subprogram or Functions in C

A C program is a collection of subprograms called functions.

A function may or may not return a value.

```
type function_name (formal parameter definition)
{
    variable declarations
    code
}
```

*C:\Users\Donna\Downloads\MyFirstCProgram.c - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

MyFirstCProgram.c MyFirstCProgram-bleh.c win801ca.c

```
1 /* Donna French 1000074079 */
2
3 /* This is my first C program for CSE 1320 */
4
5 #include <stdio.h>
6
7 int main(void)
8 {
9     printf("Hello World");
10
11     return 0;
12 }
13
```

length : 165 Lines : 13 Ln : 11 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Type here to search

11:11 AM 1/16/2019



Comments in C

Comments in C programs are not executed by the compiler – they are ignored.

```
/* This is a comment */
```

When using /* */, comments cannot be nested

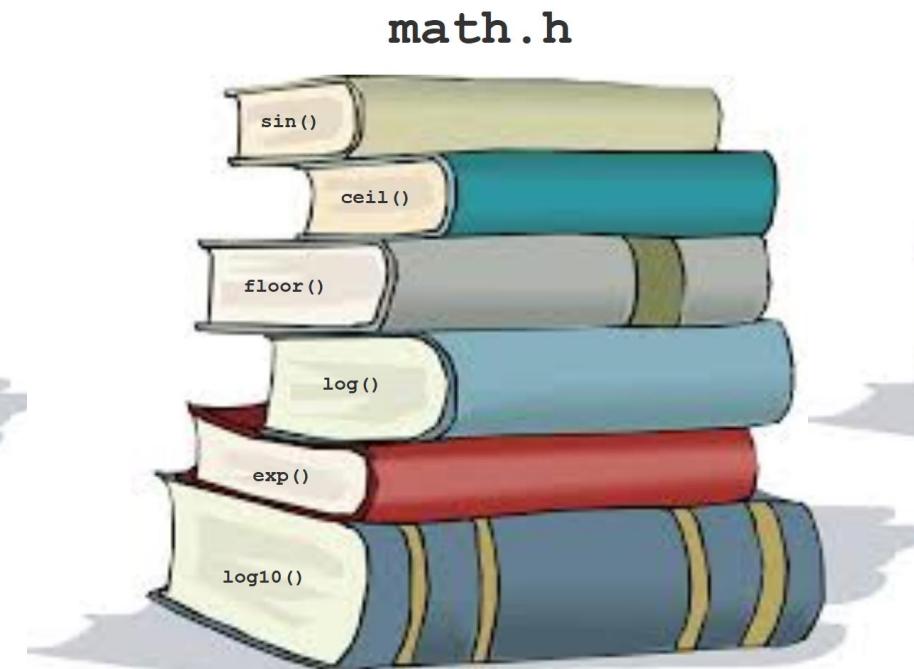
```
/* This is /* not a comment */ */
```

```
// is also a valid method of commenting in C (comes from C++)
```

```
1  /* Donna French 1000074079 */
2
3  /* This is my first C program for CSE 1320 */
4
5  #include <stdio.h>
6
7  void GoodbyeWorld(void)
8  {
9      printf("Goodbye World");
10 }
11
12 int main(void)
13 {
14     printf("Hello World");
15
16     GoodbyeWorld();
17
18     return 0;
19 }
```

Libraries in C

- A library in C is a collection of functions and definitions
- C contains many standard libraries



```
1 /* Donna French 1000074079 */
2
3 /* This is my first C program for CSE 1320 */
4
5 #include <stdio.h>
6
7 void GoodbyeWorld(void)
8 {
9     printf("Goodbye World");
10}
11
12 int main(void)
13 {
14     printf("Hello World");
15
16     GoodbyeWorld();
17
18     return 0;
19}
```





An Introduction to Output in C

Library function `printf()` will **print** formatted output.

```
printf( control_string, args, ... )
```

displays the `control_string` with conversion specifications replaced with requested conversions of the arguments.

```
printf("Hello World\n");
```

```
printf("Hello "
      "World\n");
```

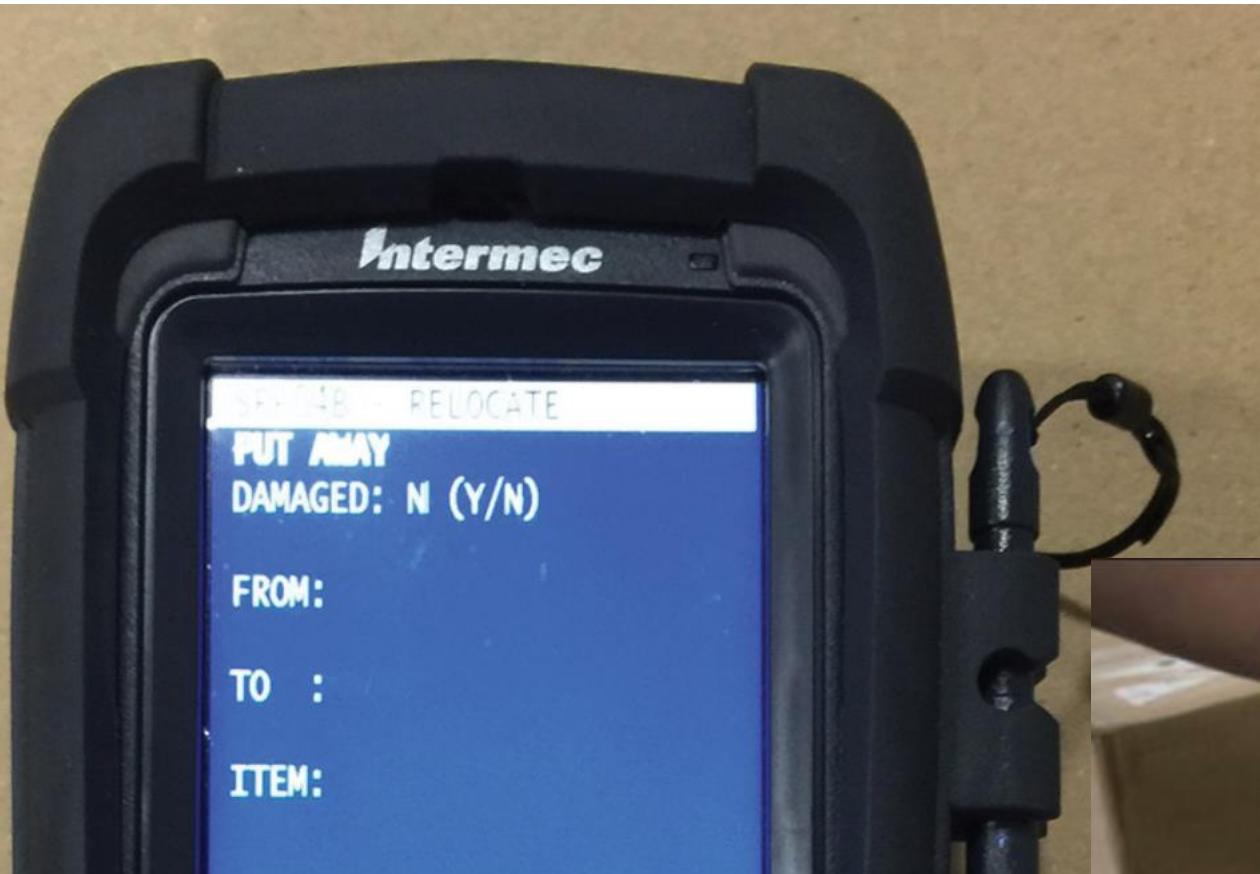


MyFirstCProgram.c

```
1  /* Donna French 1000074079 */  
2  
3  /* This is my first C program for CSE 1320 */  
4  
5  #include <stdio.h>  
6  
7  void GoodbyeWorld(void)  
8  {  
9      printf("Goodbye World");  
10 }  
11  
12 int main(void)  
13 {  
14     printf("Hello World");  
15  
16     GoodbyeWorld();  
17  
18     return 0;  
19 }  
20
```



An Introduction to Output in C



An Introduction to Output in C

Other Escape Characters like \n

- \t the tab character
- \b the backspace character
- \\" the double quote character in a string
- \' the single quote character
- \\" the backslash character
- \0 the null character

Input and Output with Variables

Variables are used to hold data while a program is executing.

- Variable must be declared before it can be used
- Declaration establishes the variable's name and type
- Compiler reserves space in memory for each variable



Input and Output with Variables

int

- built-in data type used to represent integers
- declaring a variable of type `int` with the name `counter`

```
int counter;
```

- To assign a value to `counter`

```
counter = 1;
```

- A variable can be declared and assigned a value in one statement

```
int counter = 1;
```

Input and Output with Variables

```
#include <stdio.h>

int main (void)
{
    int first, second, third = 789;
    int counter = 123;

    printf("The value of counter is %d\n", counter);

    counter = 456;

    printf("The value of first %d, second %d, third is %d\n",
           first, second, third);

    first = second = 567;
    third = first;

    printf("The value of first %d, second %d, third is %d\n",
           first, second, third);

    return 0;
}
```

Declaration with assignment

Assignment only



Input and Output with Variables

Function `scanf()` **scans** for **formatted input**.

```
scanf( control_string, args, ... )
```

scans input for characters requested by conversion specifications in the control_string. For each conversion specification in the control_string, attempts to convert the input value and store in the corresponding argument.

```
printf("Enter your favorite number\n");
scanf("%d", &FavNum);
printf("Your favorite number is %d\n", FavNum);
```

Input and Output with Variables

```
#include <stdio.h>

int main (void)
{
    int FavNum;

    printf("Enter your favorite number\n");
    scanf("%d", &FavNum);
    printf("Your favorite number is %d\n", FavNum);

    return 0;
}
```

Decimal vs Hexadecimal vs Octal

Why do programmers always mix up Halloween and Christmas?

Because Oct 31 = Dec 25

Great video explaining decimal, hexadecimal, octal and binary

<https://www.youtube.com/watch?v=5sS7w-CMHkU>

Arithmetic Operators

| | |
|-----------------------------|---|
| addition and unary plus | + |
| unary minus and subtraction | - |
| multiplication | * |
| division | / |
| remainder | % |

- Integer division
 - Truncating division
 - Fractional part (remainder) is truncated
- Remainder
 - Returns the remainder from division
- Unary + and -

Arithmetic Operators

Precedence of Arithmetic Operations

High precedence

- Unary operators $+, -$
- Multiplicative Operators $*, /, \%$
- Additive Operators $+, -$
- Assignment Operators $=$

Low precedence

Within each group, the operations associate from left to right.

$$a * b / c = (a * b) / c$$

$$2 * a + 4 / b = (2 * a) + (4 / b)$$

Action Items

- If you do not have an omega.uta.edu account, please request one
 - <https://www.uta.edu/oit/cs/web/student-faq.php#N1027F>
 - **Question: How do I get an account on omega.uta.edu?**
 - You can fill out the account request form online or go by the Help Desk on the first floor of the Library with a photo ID.
- PC users - Download FileZilla
 - <https://www.uta.edu/oit/cs/files/sftp/putty/filezilla.php>
- PC users - Download PuTTY
 - <https://www.uta.edu/oit/cs/files/sftp/putty/putty.php>
- Download Notepad++ (if needed)
 - <https://www.notepad-plus-plus.org>
- Mac instructions for ssh
 - See Blackboard
- Create HelloWorld.c and compile and run on Omega.

First homework will be posted on Monday. You will want to code some of the questions so be prepared by making sure you can access and compile on Omega.