Contact Information

Instructors
Dr. Eric Anson  eanson@email.arizona.edu
GS 823; Office Hours: Mon 11-12, Wed 2-3, Thur 1-2

Russ Lewis  russelll@cs.arizona.edu
GS 850; Office Hours: MWF 10-11

email: cs352f16@cs.arizon.edu
(This email reaches all of us)

TAs
Mallory Walsh  walshmallory@email.arizona.edu
Mai Eliza Moua  memoua@email.arizona.edu
Navdeep Singh  navdeep@email.arizona.edu
Mathias Gibbens  gibmat@email.arizona.edu

Office Hours: see web page
Two Lectures

• Section 2:  (Mr. Lewis)
  • 1-2 pm MWF    ILC 150
• Section 1:  (Dr. Anson)
  • 4-5 pm MWF    Chavez 110

• Will cover the same material each day
• You can attend either or both (except on exam days)
Websites:

http://www.cs.arizona.edu/classes/cs352/fall16/
- Class homepage

https://piazza.com/arizona/spring2016/csc352/home
- Questions and Announcements

http://d2l.Arizona.edu
- Grades

Mr. Lewis records his lectures via Panopto
- Access through “Content” tab on D2L
• Dr. Eric Anson
• Joined Computer Science Faculty Last Year
• BS Math/Comp Sci – Pepperdine University – 1985
• MS Mathematics – University of Arizona – 1993
• PhD Comp Sci – University of Arizona – 2000
• Worked all throughout industry and government from the Feds to the County and from biotech to Raytheon
• Taught 352, 245 and 473 last year
• Taught 14 math classes while a TA in mathematics
The challenge with 352

Our CS program primarily uses Java in the 300's and below, but C in many 400's.

In 352 we try to...

(1) Transform Java programmers into C programmers.
(2) Teach user-level UNIX facilities for programmers.

If you don’t learn C, you get killed in the 400's!

Therefore:

• Our priority is C programming.
• We'll focus on UNIX topics that are most directly useful to C programmers.
Course Topics

Unix

- Basic concepts
- Filesystem navigation
- File management
- Basic Commands (grep, sort, wc, head, etc)
- I/O Redirection
- shell scripts

- make
- gdb
- gcov
- valgrind
Course Topics, continued

The C programming language

- elementary C programming
- arrays, strings, pointers
- dynamic memory allocation
- structs
- file I/O
- implementing data structures
- performance tuning
Textbooks...

• No required text books . . . but

• *C Programming: A Modern Approach*, 2*nd* edition by K. N. King is an excellent book and is listed as "Recommended."
  
  • It is a good book, but expensive. You will probably want some C book as reference.

• *Learning the Bash Shell* is also recommended and is available as a free ebook from the library
Assignments

• Weekly Programming Assignments
  • Usually due on Wednesdays
  • Use turnin
  • No late work
  • Graded assignments back in 6 days (normally)
• Mostly C
• Some shell
• First one Due Aug 31
Exams

• Midterm
  • Wed, Oct 12
  • Must go to your assigned room

• Final Exam
  • Section 1 (4pm): Tue, Dec 13
  • Section 2 (1pm): Mon, Dec 12
Grading

• Assignments 60% (Drop lowest)
• Quizzes 5% (Drop lowest 2)
• Mid-term 15%
• Final 20%

Final Grade

Standard 90/80/70/60 scale

Might lower the bar, never raise it

No extra credit
Capsule summary:

Don't cheat in my class!
Don't make it easy for anybody else to cheat!

1\textsuperscript{st} Offense will result in at least a drop in letter grade
2\textsuperscript{nd} Offense will result in failing the course

Cheating for this class is passing off anyone else's work as your own or helping another student to cheat. Helping another student to cheat might mean leaving your code where someone else can get at it.
Cheating Summary

• Your integrity is worth more than a score on an assignment/test or even grade in a course

• Eventually it will catch up with you. If you don’t know the material from this class you will not make it through the classes that follow.

• The department takes cheating VERY seriously and if you get caught the consequences are grim.
Attendance

Attendance is expected but not recorded.

You will be responsible for material covered in class even if it does not appear in the notes posted online.

You've paid for the class, you should come to the class.

You probably won't do well if you're not coming to the class and surely won't do as well as you would if you did come to class.
Classroom Behavior

• The basic rule is be considerate of your fellow students (and also your hardworking instructor).

• I don’t mind use of laptops or tablets, but for those who do find them distracting let’s make the left side (my right, your left) of the classroom a “screen free” zone.

• Sit close

• Read the department code of conduct.
Be Prepared!

• This is a tough class and it takes a lot of time.

• Last year less than ½ of the students who originally enrolled in this class finished with a C or better.

• You CAN do this though. And it can be fun. But DON’T think you can do it without working.
Tips

1. Come to the class!
Tips

1. Come to the class!

2. Start Assignments Right Away!
Tips

1. Come to the class!

2. Start Assignments Right Away!

3. Come to class!
Tips

1. Come to the class!
2. Start Assignments Right Away!
3. Come to class!
4. Ask questions.
Tips

1. Come to the class!

2. Start Assignments Right Away!

3. Come to class!

4. Ask questions.

5. Play and experiment.
Tips

1. Come to the class!

2. Start Assignments Right Away!

3. Come to class!

4. Ask questions.

5. Play and experiment.

6. The assignments get progressively harder, but all have the same weight, so don’t blow off the early assignments.
1. Learn the Material!

2. Be Considerate and Kind to Others

3. Come to Class

4. HAVE FUN!!!!
We'll be using the Linux machine named "lectura" for much of our work.

By virtue of being enrolled in this class you should already have a CS computing account with the same name as your UA NetID.

If you've forgotten the password for your CS account, use "Reset my forgotten Unix password" on the page http://cs.arizona.edu/computing/services to reset it.

Note that there's no connection between your NetID password and your password on lectura.
• Since your programs will be turned in and tested on lectura, and since the Unix commands we will be using all work on lectura you need:

1. A way to connect to lectura

2. A way to either create files on or transfer files to lectura
The Mac OS is now built on Unix, so getting to lectura from a Mac would be the same as getting there from a linux system.
If you're on a Mac, start Terminal and use `ssh` to login to lectura:

1. **bash** prompt on Mac
2. `ssh yourNetID@lec.cs.arizona.edu`
3. No echo/feedback while typing password
4. **bash** prompt on lectura!

**NOTE:** Your **bash** prompts may differ!
"PuTTY" is a free Telnet/SSH client that I recommend for connecting to *lectura* from a Windows machine.

If you Google for "putty", the first hit should be this:

PuTTY Download Page

- [www.putty.org](http://www.putty.org)

Download *putty.exe*:

*putty.exe* is just an executable file; there's no installer. Save *putty.exe* to a convenient place, perhaps your Desktop.
Click on `putty.exe` to run it. In the dialog that opens, fill in `lec.cs.arizona.edu` for Host Name and click Open.
• These are ways to open a connection to lectura
• Commands you type in are sent to the shell on lectura, lectura processes them, and the results are sent back to your computer.
• You can make a connection, learn the Unix commands, use an editor like vi or emacs to create and edit files, and that's all you need . . . but
• What if you want to develop on your own computer?
Virtual Machine

• On the class web page there are links and instructions for downloading and installing a VM that mimics lectura (mostly) on your computer.

http://www.cs.arizona.edu/classes/cs352/fall16/

• A VM will let you develop code and play with commands locally, and then you can copy your files to lectura after they are finished.
Your First Tasks

• ASAP you will want to establish a connection with lectura
• You might want to download and install the VM and start playing with it.
• In the first assignment you must turn in a text file answering UNIX questions on lectura.
• You can either do this by creating the file on lectura, or copying the file to lectura.