http://www.cse.psu.edu/~deh25/cmpsc311/References.html
CMPSC 311, Introduction to Systems Programming

References

Textbooks, both required

- **Computer Systems: A Programmer's Perspective**, Randal E. Bryant, David R. O'Hallaron, second edition, 2011, Prentice Hall (Pearson), ISBN 978-0-13-610804-7. Web site, http://csapp.cs.cmu.edu
- **C Programming: A Modern Approach**, K.N. King, second edition, 2008, W.W. Norton, ISBN 978-0-393-97950-3. Web site, http://knking.com

Textbooks, previously required

- **Advanced Programming in the UNIX Environment**, W. Richard Stevens, Stephen A. Rago, second edition, 2005, Addison-Wesley, ISBN 0-201-43307-9 (hardcover), 0-321-52594-9 (paperback). Web site, http://www.apuebook.com/
- **C: A Reference Manual**, Samuel P. Harbison III, Guy L. Steele, Jr., fifth edition, 2002, Prentice-Hall, ISBN 0-13-089592X. Web site, http://www.CAReferenceManual.com/

Textbook, optional but highly recommended

- **Harley Hahn's Guide to Unix and Linux**, Harley Hahn, 2009, McGraw-Hill, ISBN 978-0-07-313361-4. Website, http://www.harley.com/books/sg3.html

Additional references

- David A. Patterson and John L. Hennessy, Computer Organization and Design, Morgan Kaufmann, fourth ed., 2009.
- Kay A. Robbins and Steven Robbins, Unix Systems Programming, Prentice-Hall, 2003.

- Bruce Molay, Understanding Unix/Linux Programming, Prentice-Hall, 2003.
- Eric S. Raymond, The Art of Unix Programming, Addison-Wesley, 2004.
    - Biculturalism, Joel Spolsky [read this first]
- Rich Teer, Solaris Systems Programming, Prentice-Hall, 2005.
- Mark Dalrymple and Aaron Hillegass, Advanced Mac OS X Programming, Big Nerd Ranch, 2005.
- Robert C. Seacord, Secure Coding in C and C++, Addison-Wesley, 2006.
- Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, second ed., Prentice Hall, 1988.
    - "This is the weapon of a Jedi Knight. Not as clumsy or as random as a blaster, but an elegant weapon for a more civilized age."

General references on using Unix

- See Hahn, above
- Paul Love, Joe Merlino, etc., Beginning Unix, Wiley, 2005.
- See also Robbins, on the Safari list, below

General references on programming and systems

- Steve McConnell, Code Complete: A Practical Handbook of Software Construction, Microsoft Press; second ed., 2004.
- Randall Hyde, Write Great Code: Volume 1: Understanding the Machine, No Starch Press, 2004.
- Randall Hyde, Write Great Code, Volume 2: Thinking Low-Level, Writing High-Level, No Starch Press, 2006.
    - Sample chapter (PDF), Chapter 8: Variables in a High-Level Language.
- Jon Stokes, Inside the Machine: An Illustrated Introduction to Microprocessors and Computer Architecture, No Starch Press, 2006.
- Norman Matloff and Peter Jay Salzman, The Art of Debugging with GDB, DDD, and Eclipse, No Starch Press, 2008.
- Michael Kerrisk, The Linux Programming Interface, No Starch Press, 2010. Looks good, but expensive.

Assorted topics you should understand already

- How To Study, by William J. Rapaport.

- What Have You Tried?, by Matt Gemmell. Good advice for students. Maybe not you, of course, but you know someone who needs it. The comments are also worth reading!
- On Integrity and Learning C#, by David Morton. A similar discussion, worth your attention.
- Peter Hosey, Everything you need to know about pointers in C.
- Everything you ever wanted to know about C types, by Peter Seebach, from IBM developerWorks
  1. What's in a type?
  2. Floating point and derived types
  3. Implementation details
  4. Portability and pitfalls

More advanced topics

- W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, UNIX Network Programming, The Sockets Networking API, volume 1, third ed., Addison-Wesley, 2004.
- W. Richard Stevens, UNIX Network Programming, Interprocess Communications, volume 2, second ed., Prentice Hall, 1999.
- Andreas Zeller, Why Programs Fail: A Guide to Systematic Debugging, Morgan Kaufmann, second ed., 2009.

Safari Books Online (free access from a PSU site, or via the PSU library)

- Stephen Prata, C Primer Plus, fifth edition, Sams, 2004.
- Debra Cameron; James Elliott; Marc Loy, Learning GNU Emacs, third edition, O'Reilly, 2004.
- Cameron Newham, Learning the `bash` Shell, third edition, O'Reilly, 2005.
- Robert Mecklenburg, Managing Projects with GNU make, third edition, O'Reilly, 2004.
- Jeffrey E. F. Friedl, Mastering Regular Expressions, third edition, O'Reilly, 2006.
- Mark Lutz, Programming Python, third edition, O'Reilly, 2006.
- Tom Christiansen, Jon Orwant and Larry Wall, Programming Perl, third edition, O'Reilly, 2000.
- Arnold Robbins, Unix in a Nutshell, fourth edition, O'Reilly, 2005. Generally useful, but sometimes it's no better than a man page.

- Ellen Siever, Aaron Weber, Stephen Figgins, Robert Love, Arnold Robbins, Linux in a Nutshell, fifth edition, O'Reilly, 2005.
- Henry S. Warren, Jr., Hackers Delight, Addison Wesley Professional, 2002. This is an amazing and wonderful book, but not for the faint-of-heart.
- Robert Love, Linux System Programming: Talking Directly to the Kernel and C Library, O'Reilly, 2007. (Try this one instead.)
- Daniel P. Bovet, Marco Cesati, Understanding the Linux Kernel, third edition, O'Reilly, 2005. Belongs in the More Advanced list, but it's a really good book.
- Mark G. Graff and Kenneth R. van Wyk, Secure Coding: Principles & Practices, O'Reilly, 2003.
- Ben Collins-Sussman, Brian W. Fitzpatrick and C. Michael Pilato, Version Control with Subversion, O'Reilly, second ed., 2008.
- Ellie Quigley, UNIX Shells by Example, fourth edition, Prentice Hall, 2004.

Wikipedia (these pages were trustworthy, last time we checked)

- C (programming language)
- The C Programming Language (book)
- System Programming
- Integrated development environment
- Single UNIX Specification
- Linux
- ELF, Extensible Linking Format, Executable and Linkable Format, Executable and Linking Format

Android

- http://developer.android.com/
- Satya Komatineni, Dave MacLean and Sayed Y. Hashimi, Pro Android 3: A Developers Handbook, Apress, 2011. This link provides a free download of Chapter 1, Introducing the Android Computing Platform, which is recommended reading.

Miscellaneous

- Unix History, complete timeline and many links
- Inside the High Tech Hunt for a Missing Silicon Valley Legend,

Wired Magazine, 2007.

- Usenix.
- American National Standard Dictionary of Information Technology
- InterNational Committee for Information Technology Standards
- ECMA, European association for standardizing information and communication systems.

Some useful reading

- Joel Spolsky, from Joel on Software. These appear to be Windows-Centric, because that's how he makes a living, but the ideas are universal; start with Biculturalism if you don't see that.
  - Advice for Computer Science College Students.
  - Back to Basics. Why you need to understand C, especially if you program in Java.
  - The Absolute Minimum Every Software Developer Absolutely, Positively Must Know About Unicode and Character Sets (No Excuses!).
  - The Joel Test: 12 Steps to Better Code.
  - Craftsmanship.
  - Biculturalism. Windows vs. Unix.
  - Martian Headsets. Web Standards.
  - Why are the Microsoft Office file formats so complicated?.
  - Talk at Yale. From Nov. 2007; 16 years after graduation, what was important?
  - The Duct Tape Programmer.
  - Making Wrong Code Look Wrong.
  - The Law of Leaky Abstractions.
  - Good Software Takes Ten Years. Get Used To It.
  - Human Task Switches Considered Harmful
  - The Guerrilla Guide to Interviewing
    - an older version
- Raymond Chen, The Old New Thing. Works for Microsoft, but don't let that stop you.
  - Cleaner, more elegant, and harder to recognize.
  - Cleaner, more elegant, and wrong.
  - When people ask for security holes as features: Privileged execution.

- Peter Norvig. Director of Research, Google.
    - Teach Yourself Programming in Ten Years.
    - The Gettysburg Powerpoint Presentation. Just about the funniest thing ever.
- Writing Guidelines for Engineering and Science Students
- Standards and specs, by Peter Seebach, from IBM developerWorks
    - Standards
    - The nitty-gritty on the C committee
    - An unsung hero: The hardworking ELF
    - Not by UNIX alone
- Real men program in C, by Michael Barr, from Embedded.com
- How Not To Optimize, By David Chisnall, from informIT (Addison-Wesley)

Some time wasters

- Mac vs. PC vs. Linux (video, 1:17; alas, a defunct link)
- Introducing the Book (video, 2:25)
- Cookie Monster vs. Coffee Break Machine (video, 4:09)
- Large Hadron Rap (video, 4:49)
- Dilbert: The Knack (video, 1:44)

Last revised, 12 Jan. 2012