

# EECS 325/425: Computer Networks

## Fall 2011

### Class Meetings:

- Tue, Thu 10:00–11:15
- Nord 410

### Instructor:

- Mark Allman
- Email: [mark.allman@case.edu](mailto:mark.allman@case.edu), [mallman@icir.org](mailto:mallman@icir.org)  
(Include “eecs325” in the Subject.)
- Office: Olin 610
- Office Hours:  
12:30–1:30 Tue, 2:00–3:00 Thu  
Other times by appointment (on campus Tue/Thu)  
Via email anytime

### Teaching Assistants:

- Will be announced in class and on the course web page as soon as they are assigned.

### Books:

- Required: *Computer Networking, A Top-Down Approach*, James F. Kurose and Keith W. Ross, 5<sup>th</sup> edition, Addison Wesley, 2009.
- Reference: *Problem Solving with C++*, Walter Savitch, 8<sup>th</sup> edition, Addison-Wesley, 2012.
- Reference: *C Programming Language*, Brian W. Kernighan and Dennis M. Ritchie, 2<sup>nd</sup> edition, Prentice Hall, 1988.

### Class Materials:

- Class web page: for slides, assignments, etc.  
URL: <http://vorlon.case.edu/~mxa99/eecs325/>  
Username: eeecs325, Password: OlinHall
- Mailing list: for announcements, answers to questions, etc.  
URL: <https://lists.case.edu/>  
Join: eeecs325-fall11  
*I will assume everyone is on this list by class on Thu, Sep/1.*
- Blackboard: for assignment submission

### Survey:

- Please take the survey at the following URL before the second class:  
<http://www.surveymonkey.com/s/CRCBDY6>

### Grading:

- Scale: 90–100 A, 80–89 B, 70–79 C, 60–69 D, < 60 F  
(curve possible at my discretion)
- Components:
  - 20% Mid-term exam
  - 25% Final exam
  - 50% Projects and homeworks
  - 5% Participation and quizzes
- There will be *no makeup quizzes*.
- ***You must turn in all projects to pass the class.***
- Graduate students will be assigned longer projects and more exam questions than undergraduates.

### Late Policy:

- Projects and homeworks are due at the *beginning of class* on the due date.
- Late penalties:
  - 10% grade reduction: up to 24 hours late
  - 25% grade reduction: 24–48 hours late
  - 50% grade reduction: > 48 hours late

### Academic Conduct:

- *All work that is turned in must be your own*, including program formulation, writing, debugging, testing and output.
- You are encouraged to liberally discuss lecture material, readings, and items that have already been turned in by all involved. You may also help each other understand the general ideas of assignments, but you are not to explicitly reveal algorithms or code, but rather derive these independently.
- Both copying and letting others copy will be considered infringements. This applies to similar assignments completed by previous students, web resources, books, etc.
- A first offense will be failure in the assignment/test in question. Continued offenses will result in course failure.
- All academic integrity violations will be dealt with according to the University's Academic Integrity Standards:  
<http://www.case.edu/provost/ugstudies/acintegrity.htm>
- Other CWRU policies can be found at:  
<http://www.case.edu/provost/ugstudies/instrman/highlights.html>

# Programming Project Guidelines

## Guidelines for all projects:

- All projects *must* be completed in C or C++.
- All submitted projects *must* compile and run on the departmental Unix facilities.
- All projects *must* include a Makefile.
- All projects *must* conform to the guidelines given in the assignment (e.g., in terms of executable name, output format, command-line options, etc.) to ensure we will be able to automate portions of the grading.
- You will not be graded on programming style, per se. However, *sloppy programming will reduce your grade.*

## Good programming practices:

- Use meaningful variable names.
- No constants inside expressions (i.e., use constant variables or `#define` macros).
- Each function/method should include at least a sentence of documentation in a comment as to its purpose.  
(Do this during the process and not as an afterthought.)
- Additionally, comment anything that might be a bit tricky to understand.
- Use proper modularization (functions, methods, classes, etc.). Methods should not be overly lengthy or try to do too much.
- While there is no prescribed style you must use, you should use a consistent style throughout each program (e.g., indentation, bracket placement, use of mixed caps, etc.).
- Use data structures appropriate for the given task. You will not be graded on optimality—and there is often not one *right* answer given that there are engineering tradeoffs at play—but grossly suboptimal code will result in a grade reduction.
- Do not waste memory (e.g., remember to `free()` memory you no longer need, etc.).
- Do not perform needless computations and remember computations that are used frequently.
- Ensure that your programs perform well with large input sets. Do not think “bytes and kilobytes”, think “megabytes and gigabytes”.
- Check return values and make sure your programs behave reasonably in the face of errors.

## Rough Schedule

Date	Assignments	Tests	Topics	Notes
Aug/29 Sep/01	#1 Assigned		Administravia and overview	First day of class
Sep/06 Sep/08			Link layer	
Sep/13 Sep/15	#1 Due #2 Assigned		<i>Network Layer</i>	
Sep/20 Sep/22				
Sep/27 Sep/29	#2 Due			
Oct/04 Oct/06	#3 Assigned			
Oct/11 Oct/13			<i>Transport Layer</i>	
Oct/18 Oct/20		<i>Mid-term</i>		
Oct/25 Oct/27	#3 Due			Fall Break; no class
Nov/01 Nov/03	#4 Assigned			
Nov/08 Nov/10			<i>Application Layer</i>	
Nov/15 Nov/17	#4 Due #5 Assigned			
Nov/22 Nov/24				Thanksgiving; no class
Nov/29 Dec/01				
Dec/06 Dec/08	#5 Due		<i>Security Architecture</i>	Last day of class
Dec/20		Final		12:30–3:30, location TBD

**Note:** This is a rough schedule with all items listed in *italics* being approximate.