

Class Meetings:

Tuesdays and Thursdays, 10:30 AM - 12:00 PM, IQSS, 2nd floor, room 22. The IQSS building stands opposite to William James Hall, the big white building on Kirkland St. For map consult: <http://www.map.harvard.edu>

Course Web Page:

For lecture notes, codes, general info, updates or whatever hop on to:
www.stat.harvard.edu/~ccr2005

Instructor and Contact Person:

Gopi Goswami, office: Science Center, room 612 in the 6 th floor
www.fas.harvard.edu/~goswami

Email:

For all course-related woes and concerns use: ccr2005@stat.harvard.edu
For personal problems, if any, use: goswami@stat.harvard.edu

Please **do not** email me at goswami@stat.harvard.edu for course-related stuff!

If you do that anyway, and I end up misplacing/loosing them please try again by re-sending them to ccr2005@stat.harvard.edu or see me before/after class.

Office Hours:

By appointment, email ccr2005@stat.harvard.edu to get one.

Texts and References:

There are plenty, we mention a few here and would add on to the list as we move on:

- For C:
 - “The C Programming Language (2e)” by Kernighan and Ritchie, excellent reference but not for beginners
- For C++:
 - “The C++ Programming Language (3e)” by Bjarne Stroustrup, excellent reference but not for beginners
 - “C++ How to Program (4e) / (5e)” by Deitel and Deitel, great book for beginners
- For R:

- “Modern Applied Statistics with S (4e)” by William N. Venables and Brian D. Ripley
- “An Introduction to R”, “The R language definition” and “Writing R Extensions” from the “Manuals” section of <http://www.r-project.org/>

Goal and Prerequisites:

The ultimate goal of this course is to help you write faster implementations of algorithms, which you might end up developing as a part of your research, written in R by incorporating a back-end support of either C (preferred) or C++.

So, some knowledge of R is absolutely essential. Although, I’ll touch upon some tricks to make pure R programs faster, main emphasis would be on teaching C, C++ and how to mingle these with R.

The main example we would be using throughout the course is a general purpose Metropolis-Hastings sampler (whatever that means), once you know how to mingle C, C++ and R you should be all set to implement whatever you want really.

Comments, Suggestions and Hate Mails:

Go to

<http://www.people.fas.harvard.edu/~goswami/>

and click on

So, gopi is your "Instructor"

link. This is very very important! Please regularly use this feature to keep me updated on how I am doing or just spell those cursings out loud. If you want to make me teach you better this is the way to do this.

Assignments and Grading:

I would throw in some coding problems as “assignments” at certain points in the course. They would not be due every week and they would be optional, so no worries. However, if you don’t do the coding, you won’t learn much, so it is in your best interest to do them religiously.

I would ask interested members of the audience to form groups of size two and do the coding in collaboration. For grading, we are going to pair up groups and the two groups within a pair would grade each other’s assignment.

Why do this? Well, as you would know or would come to know in the near future, reading other people’s code is a goes a long way to improve one’s programming skills.

—Happy Coding!—

Topics Covered:

The plan is to cover the following topics not necessarily in the order mentioned. Want me to cover more? Feel free to send me your suggestions. Click on the “Teaching Suggestions” link on the course web page for sending your suggestions.

- C stuff
 - Pointers, Arrays, Multidimensional Arrays
 - Control Flow
 - Functions
 - Memory manipulation
 - Structures, Structure Pointers, Function Pointers
 - String manipulation
 - Preprocessor, Macros
 - Input/Output
- C++ stuff
 - Class
 - Inheritance and Composition
 - Polymorphism
 - Templates
 - Standard Template Library interface: STL
 - Input/Output
- R stuff
 - .C interface
 - .Call interface
 - .External interface
- Other stuff
 - Basic UNIX utilities
 - Compilation tool: Makefile
 - Debugging tool: `gdb`
 - Version control system: CVS/Subversion
 - Editor: (X)Emacs

- If time permits:
 - Scripting language: Python
 - Code reading basics
 - Profiling utility: gprof