Logistics

- **Instructor**: Amit Jain  [http://cs.boisestate.edu/~amit](http://cs.boisestate.edu/~amit)
- **Class website**:  
  [http://cs.boisestate.edu/~amit/teaching/253/cs253.html](http://cs.boisestate.edu/~amit/teaching/253/cs253.html)
- **Office**: MEC 302E
- **Office hours**: Check class website
- **CS Tutoring center**: Check tutoring center website for 253 tutors and their hours:  
Grading

- Individual Programming Projects (50%)
- In-class quizzes (30%)
  - Individual quiz every two weeks
  - Team quiz following the individual quiz
  - Lecture and discussion to follow
- Final exam (20%)
Team Based Learning

- Apply
- Question
- Reflect upon material and
- Discuss content as a group
Team Based Learning

- Quiz every week unless specified otherwise
- Each quiz will be taken first as an individual
  - Same quiz will be taken as a team
  - Instant feedback for the team
- TBL quizzes — individual vs team performance
- Readings will be required outside of class
- Make teams
Build Teams

Prioritized Sorting Criteria:

- Do you have any experience programming in C? Number of years?
- Do you have any real-world software development experience (internship or career)?
- Prior experience using Linux/Unix command line and system utilities/scripting?
- Is computer science your first major?
- Have you ever lived outside of Idaho?
- Are you excited to take this class? ;-)

Meet team members and introduce yourself!
Goals

By taking this course the student will be able to:

- design and develop programs of moderate complexity in C,
- translate their knowledge of object-oriented programming in Java to C,
- use various tools like IDEs, build tools, debuggers, version control and memory checkers to improve their productivity,
- use shell commands and system utilities, and
- use basic system calls related to files, processes and threads.
Where does 253 fit?

- Prerequisite for required class: 453 (Operating Systems)
- Prerequisite for various electives:
  - 425 (Introduction to Computer Networks)
  - 430 (Parallel Computing)
  - 450 (Programming Language Translation)
  - 455 (Distributed Systems),
  - 457 (Artificial Intelligence)
So, what is systems programming?

**Systems software versus Application software**
Classify the following: *wordprocessor, spreadsheet, video game, C compiler, Java compiler, Python interpreter, bash shell, standard C library, database software, Eclipse.*

- operating systems are the quintessential systems programs
- systems programming often uses features specific to hardware devices
- systems programming often uses features and APIs specific to a given operating system
- systems programming deals with objects and concepts that are typically low-level
- however concepts from systems programming are used in application programming and vice-versa
Why C?

- C is the most widely used systems programming language (followed by Java and C++)
- C is low-level and procedural while Java is high-level and object-oriented. Knowing these two languages gives you a strong basis for learning other languages down the road
- Overall, Java and C are the two most commonly used languages in the industry.
  http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html
- Internship and job interview questions are mostly based on CS 121, 221, 253 and 321.
Major topics

- Linux (and Microsoft Windows) programming environments (1 week)
- C programming (7 weeks)
- Programming tools (2 weeks)
- Shell commands and scripts (1 week)
- Basic systems programming in Linux (and Microsoft Windows) (4 weeks)
Collaborative Learning using Piazza

“Piazza is a free online gathering place where students can ask, answer, and explore 24/7, under the guidance of their instructors.”

- Piazza invite sent out
- Use Piazza to help each other
- Ask questions anonymously
- Answer questions and doubts that everyone seems to be having
Similar to working on projects in CS 121 and CS 221

The GCC C compiler is available on the Linux machines in the labs that can be used directly from the command line

You should, however, use Eclipse with CDT to begin with

- Download from www.eclipse.org/cdt (C/C++ Development Tooling plugin) and install plugin in your personal Eclipse setup. The CDT plugin is already installed in the lab.
- Try downloading and configuring it before next class - Post questions on Piazza, drop in the tutoring center, come to office hours for help
#include <stdio.h>

int main(void)
{
    printf("Hello world!\n");
    return 0;
}
Compiling and Running

- gcc -Wall helloworld.c
  - The compiler is called **gcc**, which stands for the GNU C Compiler. It is a free, open source compiler that is widely used.
  - Creates an executable named **a.out**
  - Type `./a.out` to run the program
  - The option `-Wall` asks the compiler to provide all warnings about the code, which can save us a lot of effort later!

- gcc -Wall helloworld.c -o helloworld
  - To create an executable called **helloworld**

- Now create a C project in Eclipse and compile and run the hello world program from Eclipse
Exercises

- Read pages 5-21 of the K&R C book
- Activate your Piazza account
- Configure Eclipse C/C++ CDT plugin on your computer
- Install Linux in a virtual machine on your laptop (use VMWare Player for the virtual machine). We recommend using Fedora Linux (18 is installed in the lab but you can use a newer version).