1. **Course number and name**
   CptS 360: Systems Programming

2. **Credits and contact hours**
   4 credits, 3 lecture hours, 3 lab hours

3. **Instructor's or course coordinator's name**
   KC Wang

4. **Textbook, title, author, and year**
   K.C. Wang. 2018. *Systems Programming in Unix/Linu*x, Springer International AG.
   K.C. Wang. 2015. *Design and Implementation of the MTX Operating System*. Springer International AG.
   Other supplemental materials
   Class website: http://www.eecs.wsu.edu/~cs360

5. **Specific course information**
   a. *Catalog description:* Implementation of systems programs, concepts of computer operating systems; laboratory experience in using operating system facilities.
   b. *Prerequisites or corequisites:* CptS 223; CptS 260 or EE 234, all with C or better. Certified major in computer science, computer engineering, or electrical engineering.

6. **Specific goals for the course**
   By the end of the course, students will be able to
   - Understand the operating environment of Unix/Linux (7c, 7d, 7f, 7g)
   - Write C programs to interface with the Unix/Linux kernel (2a, 2e, 2g, 6a, 6b, 7c, 7d, 7f, 7g)
   - Develop a Linux compatible EXT2 file system (2a, 2e, 2g, 5b, 5d, 5e, 5g, 6a, 6b, 6c, 6d)

7. **Brief list of topics to be covered**
   - Introduction to the Unix/Linux Operating System: Files, directories, special files, logical organization of Unix/Linux file system; user account, login process and command execution.
   - Program development: Source files; compiler, assembler and object files; linker, library and executable files; loader and execution images. Symbolic debugger and run-time support.
   - Execution image of C programs: Code, data and stack segments; function calling convention, stack frames and parameter passing; long jumps.
   - File I/O: System calls and low-level file I/O; open, close, read, write, lseek, file descriptors and file sharing. Execution of User mode and Kernel mode images, implementation and implications of system calls.
   - File Control: Permissions and access control, fcntl, chown, chmod, hard and soft links, file status and statistics. I/O redirection, pipes, filters and applications.

- Standard I/O Library: Streams and high-level file I/O; user space buffering, relationship with low-level I/O, char and line mode I/O. Formatted I/O
- File system implementation: Inodes and file representation; mkfs and physical file system layout; traversal of the file system tree; booting system images.
- Processes: Concept and implementation of processes, process execution environment, user mode and kernel mode images, process states transitions. Processes in Unix/Linux system; init, login and user processes.
- Process Control: fork, vfork, wait, exit, kill, exec operations. traps and signal handling.
- Process Synchronization and Communication:
  Signals, pipes, semaphores, messages and shared memory segments.
- Introduction to TCP/IP and Internet; hosts, IP address, TCP/IP protocols, routing; protocols, client and servermodel; ftp, rlogin, nfs, and nis,
- Network Programming using TCP/IP, HTTP, WEB and CGI programming.
- Introduction to relational database systems: mySQL
- Project: Implement a Linux compatible ETX2 File System Simulator which supports all file operations in the Linux kernel.
- Misc Topics: Makefiles and sh programming, Unix/Linux system administration