

OFFICIAL COURSE OUTLINE & OBJECTIVES

# Course Title: CIS 275 – Systems Programming

**NOTE -** CIS 272-Data Structures , CIS 254-Computer Organization and CIS 275-Systems Programming are all courses that are part of the Computer Science Program at York County Community College. Based on student enrollment and readiness, these courses have not needed to be scheduled. The college does plan to offer these courses in the coming year.

**Course Description:** This course covers systems programming concepts and software, including the C programming language, the Unix programming environment, and operating system interfaces. Students will design and implement programs in C that directly use operating system functions such as processes, timers, signals and the file system. Implementation of common Unix commands and tools will also be covered. **Prerequisites:** CIS 254 and CIS 272.

Credit Hours: 3

Clock Hours: 3

# Course Outline:

1. Introduction to C
   1. Language syntax
   2. Conditions, loops, and other flow control
   3. Standard Input/Output (I/O)
   4. Memory allocation and pointers
   5. Implementing data structures in C
2. Programmatic View of an Operating System
   1. Application programmer interfaces (APIs)
   2. System call interface
   3. Header files
   4. Libraries
3. Interacting with the Operating System
   1. File I/O and file handles
   2. Device I/O and I/O control
   3. Directories, directory entries, and directory links
   4. Timers and alarms
4. Concurrent Programming
   1. Concurrency issues
   2. Processes and signals
   3. Traditional and named pipes
   4. Socket interface and network programming
   5. Threads
   6. Locks and semaphores

# Course Objectives:

In successfully completing this course, the students will be able to:

1. Write and debug programs in the C programming language. (I)
2. Describe the value of APIs and middleware. (II)
3. Write a program that uses file I/O to provide persistence across multiple executions. (III)
4. Describe the difference between processes and threads. (IV)
5. Use mutual exclusion to avoid a given race condition. (IV)
6. Define Unix signals and signal handlers, and describe their use. (IV)
7. Demonstrate the potential run-time problems arising from the concurrent operation of many separate tasks. (IV)
8. Write C programs that use the Unix system call interface. (II-IV)
9. Describe how software can interact with and participate in various systems including information management, process control, and communications systems. (III-IV)

This workforce product was funded by a grant awarded by the U.S. Department of Labor’s Employment and Training Administration. The product was created by the grantee and does not necessarily reflect the official position of the U.S. Department of Labor. The Department of Labor makes no guarantees, warranties, or assurances of any kind, express or implied, with respect to such information, including any information on linked sites, and including, but not limited to accuracy of the information or its completeness, timeliness, usefulness, adequacy, continued availability or ownership.

[Creative Commons License](http://creativecommons.org/licenses/by/4.0/)  
Except where otherwise noted, this work by York County Community College is licensed under the [Creative Commons Attribution 4.0 International License](http://creativecommons.org/licenses/by/4.0/).