New Syllabus of record – Approved by College Curriculum Committee and in the pipeline to be approved by the Undergraduate University Wide Curriculum Committee and the University Senate

I. Course Description

COSC 430 Introduction to Systems Programming 3c-0l-3cr

Prerequisite: Grade of C or better in COSC 300 and 310, or permission of instructor

An in-depth introduction to a systems programming, system programming language(s) and application of those language(s) to systems level problems. The focus will be on programming constructs that are closely aligned with the architecture of a digital computer including those providing portability between platforms, dynamic allocation and management of virtual memory, complex in-memory data structures, reading/writing binary data using sequential and random access, pointer arithmetic/manipulation, and interaction between threads/processes.

II. Course Outcomes

Upon successful completion of the course, the student will be able to:

1. Enumerate and explain the function of the common operating system kernel routines that are provided by an operating system and accessible from a systems programming language.
2. Design, write, and test moderately complicated low-level programs using a systems programming language.
3. Proficiently use a preprocessor to implement code that is portable between different computing platforms.
4. Implement routines that read and write structured binary files such as word processing documents, index systems, or serialized hierarchical data.
5. Use operating system kernel calls from within a programming language to allocate/free virtual memory, initiate and synchronized multiple threads/processes, interact with the file system, set and respond to timers/interrupts.
6. Implement routines that implement complex data structures which superimpose arrays, records, and references on unstructured blocks of memory.
7. Implement that exploit the use of pointers to improve efficiency.

III. Tentative Course Outline:

1. System Programming and what languages are used? 2 hrs
   - A. What is Systems Programming
   - B. Explanations of specific system features
   - C. Assembly for systems programming
   - D. Overview of high level system programming languages

2. Operating system functions 3 hrs
   - A. Device management
   - B. Memory management
   - C. Process management
   - D. File system management
   - E. Accounting and security
   - F. User services

3. Case study of a high level systems programming language 6 hrs
   - A. Data types, operators, expressions
   - B. Flow of control
   - C. Functions and program structure
   - D. Scopes

E. Pointers, arrays, structures, and unions
F. Basic input and output

4. Machine considerations and portability — 3 hrs
   A. Instruction formats/types between platforms
   B. Addressing modes and address spaces between platforms
   C. Registers between platforms
   D. Data representations between platforms
   E. Preprocessor directives and portability
   F. Macros, inline assembly, and typdefs

Exam 1 — 1.5 hrs

5. Modularization and program assembly — 2 hrs
   A. Multi-file development (interfaces, APIs, header files, make files)
   A. Libraries, archives, and shared objects
   B. Dynamic and static linking

6. Memory Management — 3 hrs
   A. Arrays, records, unions from a memory perspective
   B. Allocation and de-allocation of memory from the operating system
   C. Pointer casts, arithmetic, navigation, and field references
   D. Memory corruption issues, detection and resolution

7. Input/Output at a systems level — 3 hrs
   A. Binary input and output
   B. Sequential and random access
   C. On-disk data structures
   D. Indexes and other organizational structures

8. Device drivers — 2 hrs

9. Files systems and directories — 3 hrs
   A. File system and directory architecture
   B. Disk architecture
   C. Access and update of directory attributes
   D. File Permissions

Exam 2 — 1.5 hrs

10. Process management — 3 hrs
   A. Threads
   B. Spawning processes
   C. Sleep, wait, and nap
   D. Synchronization

11. Inter-process communication — 3 hrs
   A. Pipes
   B. Sockets
   C. Signals and signal handlers
   D. Shared memory
   E. Secure Sockets
   F. Certificates

12. Object-Oriented extensions of a systems programming language — 6 hrs

A. Class definition including constructors and destructors
B. Encapsulation, inheritance, polymorphism
C. Derived classes, abstract classes, multiple inheritance
D. Generics/templates
E. Operator overloading
F. Exception handling

Total	42 hrs
Final	2 hrs

IV. Evaluation Method

Grade distribution:

- Quizzes 10%
- Exams 30%
- Final Exam (Cumulative) 20%
- Programming Projects 40%

Grade Scale:

- 90 – 100% A
- 80 – 89% B
- 70 – 79% C
- 60 – 69% D
- < 60% F

Attendance Policy:

Attendance is crucial to success in this course. To encourage class attendance, the following policy will be used: Attendance will be taken at every class. For each unexcused absence, starting with the fourth, 2% will be deducted from the overall class grade. Generally, excused absences involve illness with a doctor's excuse, verifiable family emergencies, or conflicting university activity.

V. Textbook(s)

Kernighan, B. and Ritchie, D., *The C Programming Language*, 2nd Edition, Prentice Hall, 1998.

Stevens, R. W and Rago, R. A. *Advanced Programming in the UNIX environment*, 2nd Edition, Addison-Wesley, 2005.

VI. Special Resource Requirements

None.

VII. Bibliography

Deitel & Deitel, *C++ How to Program: Late Objects Version*, 7th Edition, Prentice Hall, 2011.

Hoover, A., *System Programming with C and Unix*, Addison-Wesley, 2009

Kerrisk, M., *The Linux Programming Interface: A Linux and UNIX System Programming Handbook*, New Starch Press, 2010.

Liang, Y. Daniel, *Introduction to Programming with C++*, 2nd Edition, Pearson, 2010.

Malik, D. S., *C++ Programming: From Problem Analysis to Program Design*, 5th Edition, Course Technology, 2010.

Robbins K., and Robbins, S., UNIX *Systems Programming: Communication, Concurrency, and Threads*, Prentice Hall, 2003.

Stroustrup, Bjarne, *Programming: Principles and Practice Using C++*, Addison-Wesley Professional, 2008.

Stroustrup, Bjarne, *The C++ Programming Language*, 3rd Edition, Addison-Wesley Professional, 1997.

Weiss, Mark A., *C++ for Java Programmers*, Pearson/Prentice Hall, 2004.