COSC 420 **HIGH PERFORMANCE COMPUTING**
(Tentative)

Description: The course will study principles, practices, and implementations of parallel and distributed computing. It covers three areas of high performance computing: modern computing architectures, algorithm design, and applications and programming. Through this course, students will not only learn fundamental concepts of high performance computing but also gain hands-on hardware and programming experience in this field.

Prerequisite: Computer Science II (COSC220), Microcomputer Organization (COSC250); each with a grade of C or better
Credits: 4 units
Required Text: *Using MPI: Portable Parallel Programming with the Message-Passing Interface (3E),* by Gropp, Lusk, and Skjellum, 2014.

References:

- "*Programming Massively Parallel Processors: A Hands-on Approach*",
D.Kirk and W. Hwu, *Morgan Kaufmann,* 2010
- "*MPICH User's Guide*", *Pavan Balaji, et al.,* 2015
(https://www.mpich.org/static/downloads/3.1.4/mpich-3.1.4-userguide.pdf)
- "*The Debian Administrator's Handbook*", Raphaël Hertzog, *Roland Mas,* 2013
(https://www.debian.org/doc/manuals/debian-handbook/)

TOPICS	***Weeks***
Introduction to High Performance Computing Limits of Sequential Computing, Concurrency and Performance Analysis Flynn's Classical Taxonomy: SISD/SIMD/MISD/MIMD	**2.0**
Parallel Processing, Memory Architecture, Modern Supercomputing Multistage Interconnection, Shared Memory: UMA/ NUMA, Distributed Memory, Hybrid Memory	**2.0**
Distributed Systems and High Performance Computing (HPC) The HPC Stack, Grid Computing, Cloud Computing, Job Scheduling, Load-Balancing	**2.0**
HPC Design and Construction Linux HPC as Supercomputing platform, HPC Stack implementation on ARM and i686 Architectures (via commodity hardware: Raspberry Pi and desktop PC), Network Engineering, Administration, Security, Monitoring	**5.0**
Parallel Programming Model and Algorithm Design Principles Programming Models and Languages; Message Passing; Data Parallel Algorithm Design, Decomposition, Dependency; Multi-Thread Programming in Python/POSIX; Parallel programming with OpenMP/MPI	**4.0**

Tests 1.0

EVALUATION

Presentations, Labs, Programs, and Projects 70%
Tests and Final Exam 30%

REC 1/2016