1. **Course Number and Course Title:**
   CMP 418 – Multicore Computing

2. **Credits hours**
   3-0-3

3. **Prerequisites and/or co-requisites**
   Prerequisites: CMP 310 Operating Systems

4. **Name and Contact Information of Instructor:**
   Dr. Gerassimos Barlas

5. **Course Description**
   Covers models of parallel computation and software development on multicore systems. Examines problem decomposition patterns including divide-and-conquer, geometric decomposition, task parallelism and pipelining. Covers program structure patterns such as master-worker, map-reduce and fork-join. Provides hands-on experience with high-performance multicore and many-core platforms. Examines state-of-the-art software tools for both Central Processing Unit and Graphics Processing Unit architectures.

6. **Textbook, title, author, and year**

Textbook:

- Gerassimos Barlas, “Multicore and GPU Programming: An Integrated Approach”, 1e, Morgan Kaufmann, ISBN-13 978-0124171374, 2014

Supplemental material:

- Ian Foster, Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering, Addison-Wesley Pub Co, ISBN: 0201575949, 1995, also available on-line at http://www.mcs.anl.gov/dbpp/
- Selected material from online sources:
  - MPI 3.1 specification
  - OpenMP 5.0 specification
  - NVidia's CUDA tutorials and reference material

7. **Course Learning Outcomes**

Upon completion of the course, students will be able to:

1. Explain Flynn's taxonomy of computer architectures.
2. Describe the most important contemporary multicore machine architectures.
3. Calculate speedup and efficiency to measure the performance of a parallel algorithm.
4. Use MPI to write distributed memory programs.
5. Transform a sequential program into a multi-threaded one using OpenMP compiler-directives.
6. Design and implement programs running on GPUs by utilizing the CUDA platform.
7. Employ different software design patterns (e.g. master-worker, pipelining, map-reduce, etc.) for developing parallel applications.

8. Teaching and Learning Methodologies:

Methods include lectures, homework, quizzes, exams, class discussions and a project.

9. Course Topics and Schedule:

Topics covered	Week
Parallel computers, Flynn’s taxonomy, modern architectures	1
Amdahl’s & Gustafson’s laws, speedup and efficiency	2
Parallel program design methodologies: master-worker, pipelining, map-reduce.	3
Shared memory programming with threads	4
Shared memory programming with threads : sample applications	5
MPI: Introduction	6
MPI: Point to Point Communications	7
MPI: Collective communications	8
MPI: master-worker, sample applications	9
Shared memory programming with OpenMP	10
OpenMP directives and clauses	11
GPU programming with CUDA: introduction, grids and blocks	12
CUDA: GPU memory hierarchy	13
CUDA: optimizations, unified memory	14
CUDA: sample applications	15
Exams and Quizzes	16