1. **Course number and name**  
   CptS 411: Introduction to Parallel Computing

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

3. **Instructor’s or course coordinator’s name**  
   Ananth Kalyanaraman

4. **Textbook, title, author, and year**  
   There is no required text. All required course materials will be made available to all students on the course website <https://eecs.wsu.edu/~ananth/CptS411/> over the course of the semester. These materials include prepared lecture notes, in-class lecture scribes, and a variety of online tutorials and reading materials identified by the instructor.  
   **Other supplemental materials**  
   The following books will serve as references (no need to purchase):  
   A. Grama, G. Karypis, V. Kumar, A. Gupta. *Introduction to Parallel Computing* (2nd ed.), Addison-Wesley.  
   P.S. Pacheco and M. Kaufman. *Introduction to Parallel Programming*.  
   P.S. Pacheco and M. Kaufman. *Parallel Programming with MPI*.  
   F.T. Leighton and M. Kaufmann. *Introduction to Parallel Algorithms and Architectures*.

5. **Specific course information**  
   a. **Catalog description:** Introduction to Parallel Computing  
   b. **Prerequisites or corequisites:** CPT S 215, 223, or 233, with a C or better; certified major in Computer Science, Computer Engineering, Data Analytics, Electrical Engineering, or Software Engineering.

6. **Specific goals for the course**  
   By the end of the course, students will be able to  
   - Demonstrate a conceptual understanding of the fundamental principles behind parallel computing, parallel programming models, and parallel algorithm design and analysis (1b, 1c, 1d, 1e, 6a).  
   - Implement parallel programs using primarily MPI (for distributed memory systems) and OpenMP (for shared memory multicore systems) (1d, 1e, 2e, 2g, 7e, 7d, 7f).  
   - Identify and differentiate among parallel architectures and network interconnect models (1b, 1e).  
   - Demonstrate proficiency in parallel algorithm design by designing parallel algorithms for a stated problem, analyze their complexities, estimate/predict performance, articulate tradeoffs, and implement algorithms in the form of computer programs (1a, 1b, 1d, 1e, 2a, 2b, 2g, 6a, 6b, 6c, 6d, 7c, 7d, 7f).
● Effectively document and communicate empirical results in a manner that is consistent with scientific practice, including providing reasoning and rationale in written documents (3a, 3b, 3c, 3d, 3e, 6b, 6c, 6d).
● Demonstrate an ability to abstract parts of a real world/scientific computing problem into a parallel computing problem, and devise solutions (1a, 1c).
● Function effectively as part of a team (5b, 5c, 5g).

7. **Brief list of topics to be covered**
   ● Fundamentals in parallel algorithm design and analysis
   ● Parallel computation models
   ● Network interconnects and embeddings
   ● MPI programming
   ● OpenMP shared memory multicore programming
   ● Parallel reduction operations
   ● Parallel prefix
   ● Sorting
   ● Matrix operations
   ● Parallel scientific computing applications
   ● Parallel graph analytics
   ● GPU Programming
   ● MapReduce and cloud computing